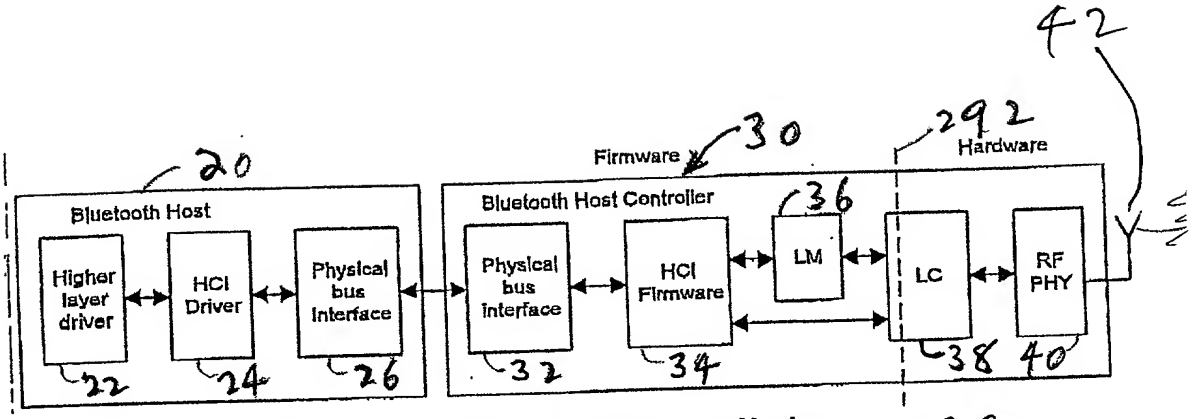


FIG. 1 PRIOR ART

Figure : Typical partition between firmware and hardware

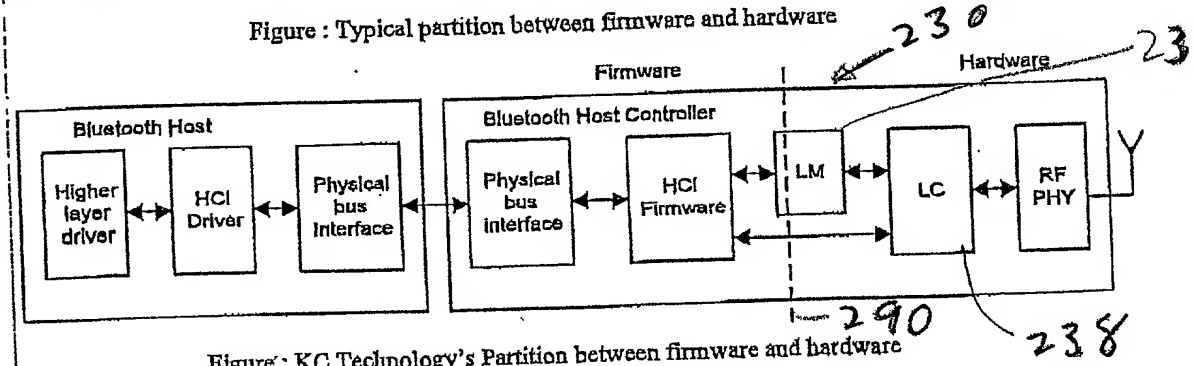


FIG. 13

Figure : KC Technology's Partition between firmware and hardware

AUG

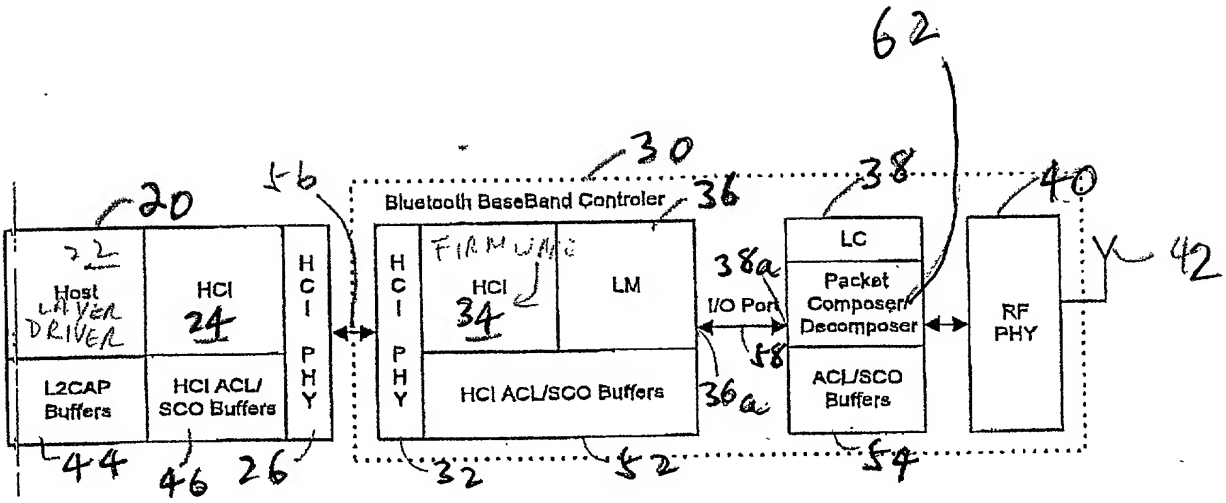


FIG. 2
PRIOR ART

Figure : HCI ACL Data Packet

31 ... 24	23 ... 16	15 ... 8	7 ... 0
Data Total Length		BC / PB	Connection Handle
...		Data byte 1	Data byte 0

FIG. 3A
PRIOR ART

Figure : HCI SCO Data Packet

31 ... 24	23 ... 16	15 ... 8	7 ... 0
Data byte 0	Data Total Length	Reserved	Connection Handle
		...	Data byte 1

FIG. 3B
PRIOR ART

FIG. 4 PRIOR ART

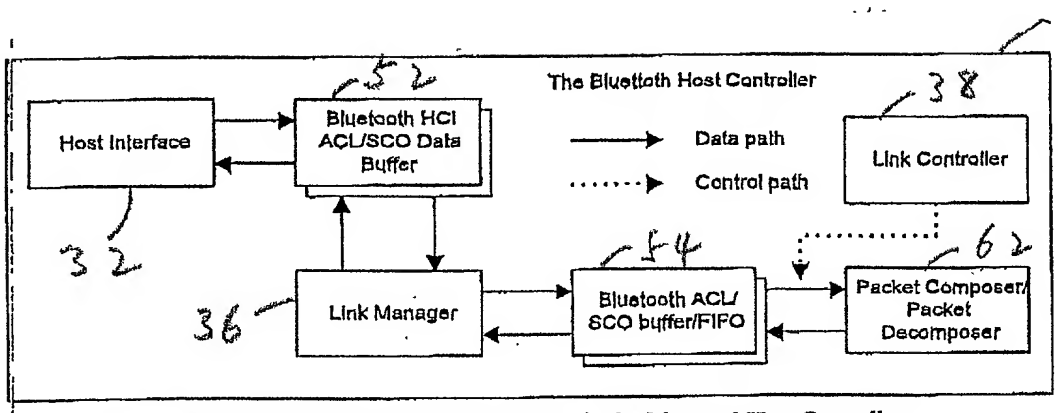


Figure : Data flow and Buffer scheme in the BluetoothHost Controller

FIG. 5A PRIOR ART

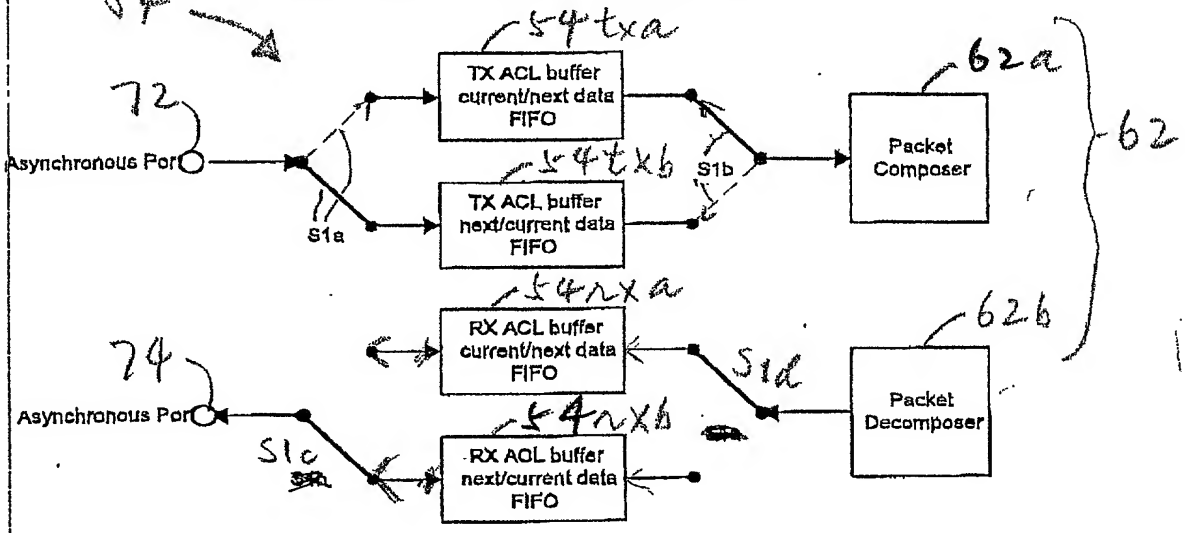
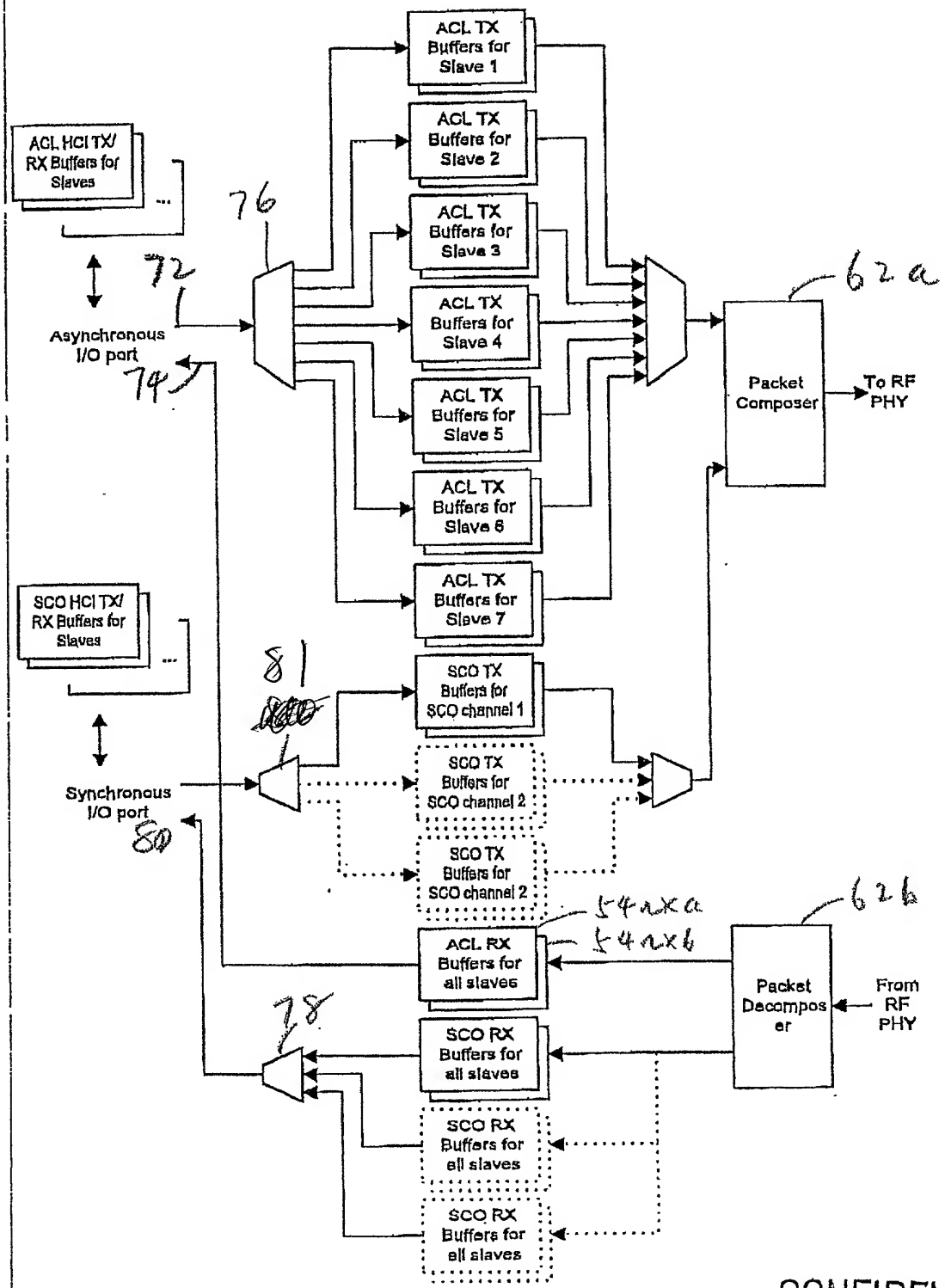


Figure : Dual buffer scheme for ACL packet transmission



CONFIDENTIAL

FIG. 5B PRIOR ART

FIG. 6
PRIOR ART

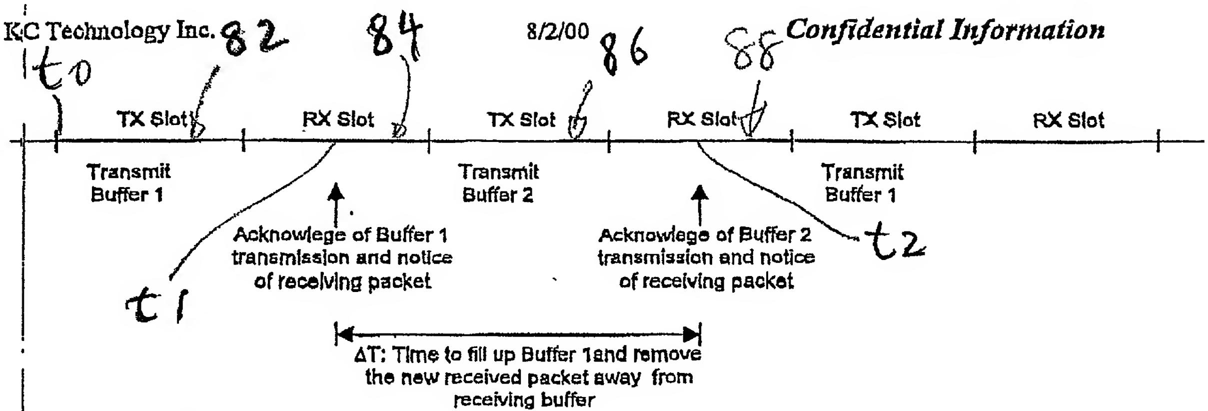


Figure : The worst case timing (ΔT) for the LM to load a TX buffer and unload a RX buffer

CONFIDENTIAL

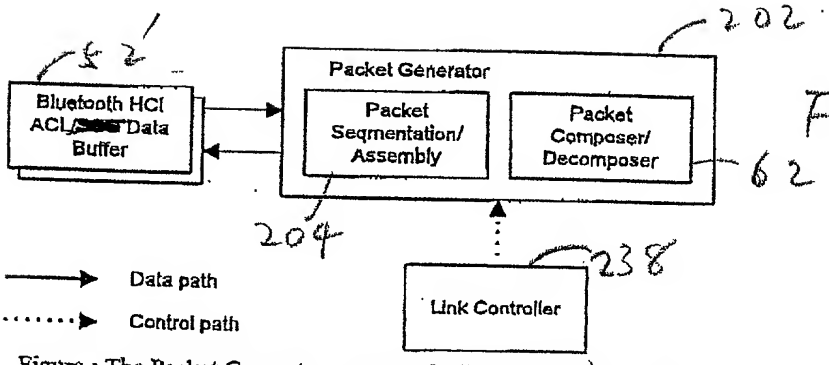


FIG. 7

Figure : The Packet Generator accesses the HCI ACL/SCO buffers directly

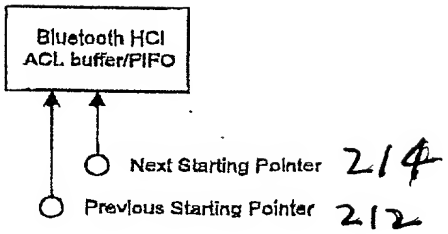
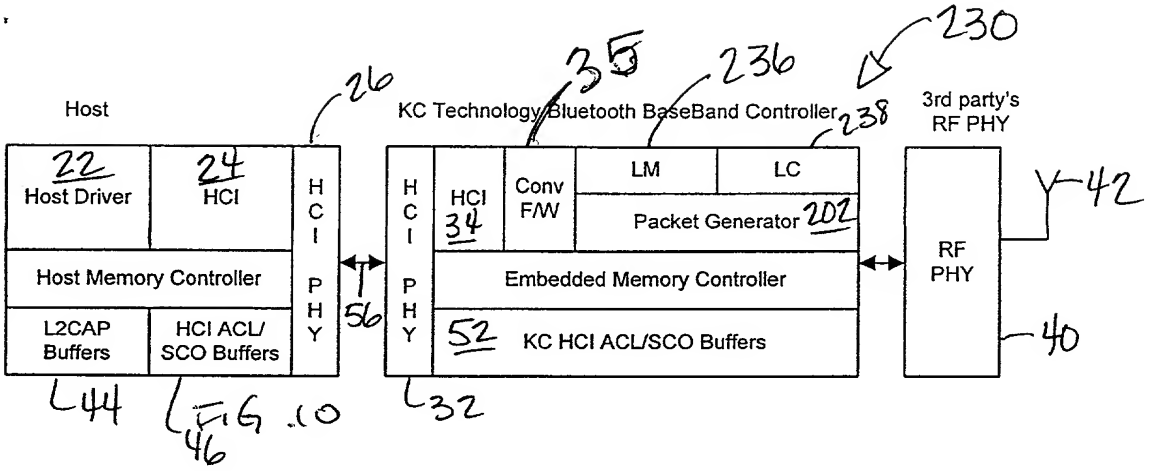

Bluetooth HCI
ACL buffer/FIFO
Next Starting Pointer
214
Previous Starting Pointer
212

FIG. 8



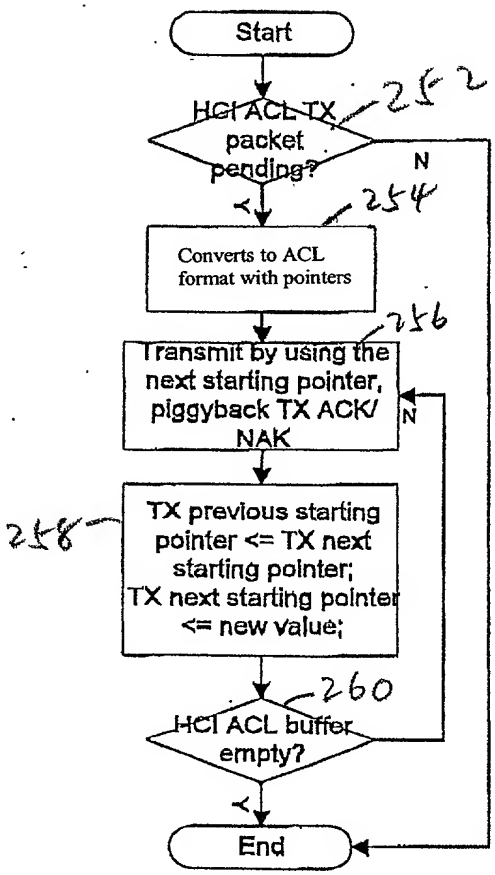


Figure : ACL transmission control flow of KC Technology's partition

FIG. 11

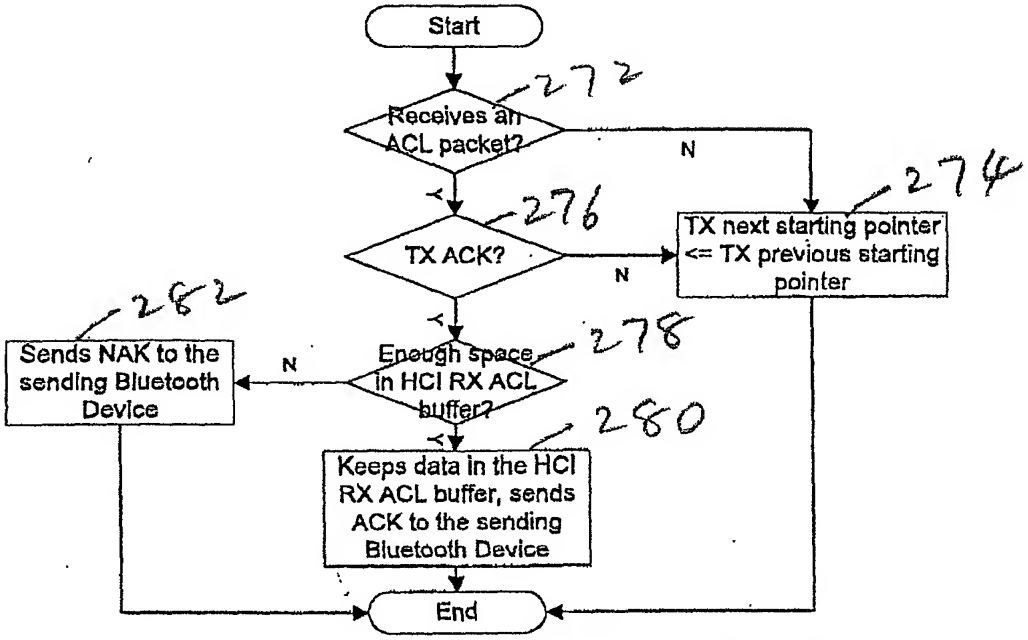


Figure : ACL receiving control flow of KC Technology's partition

FIG. 12

FIG. 14A

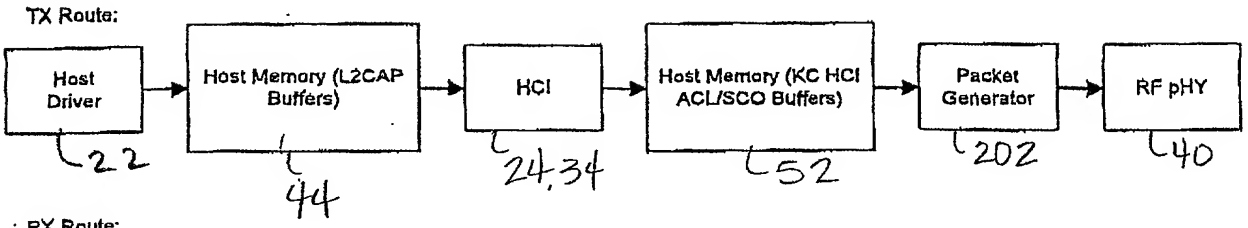
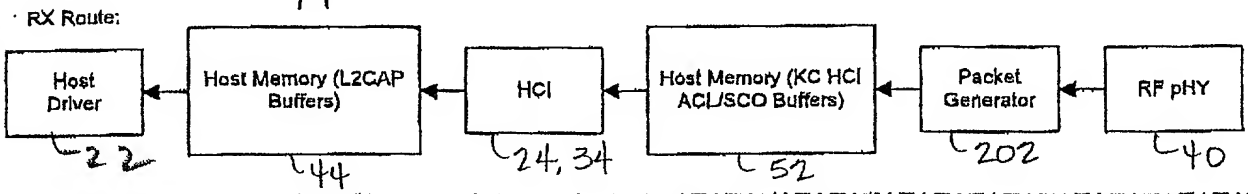


FIG. 14B



Type: A

FIG. 15C

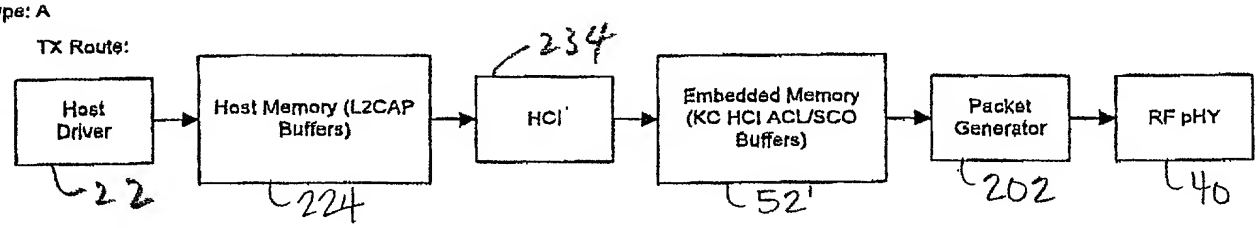
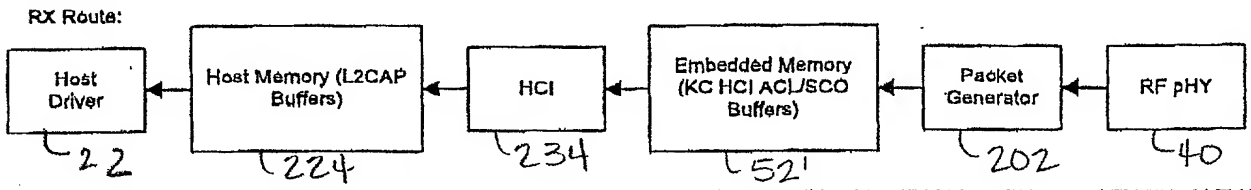


FIG. 15D



Type: B

FIG. 17C

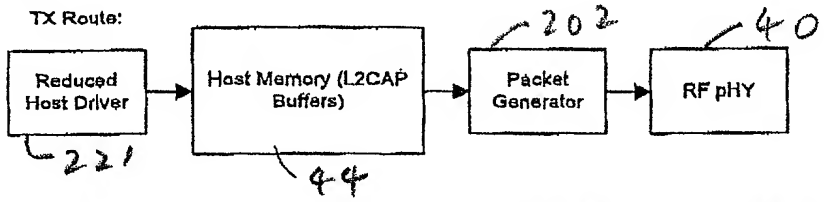
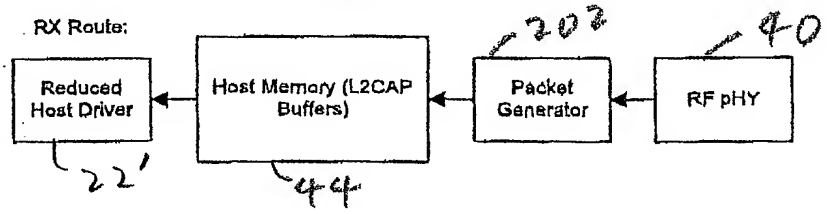


FIG. 17D



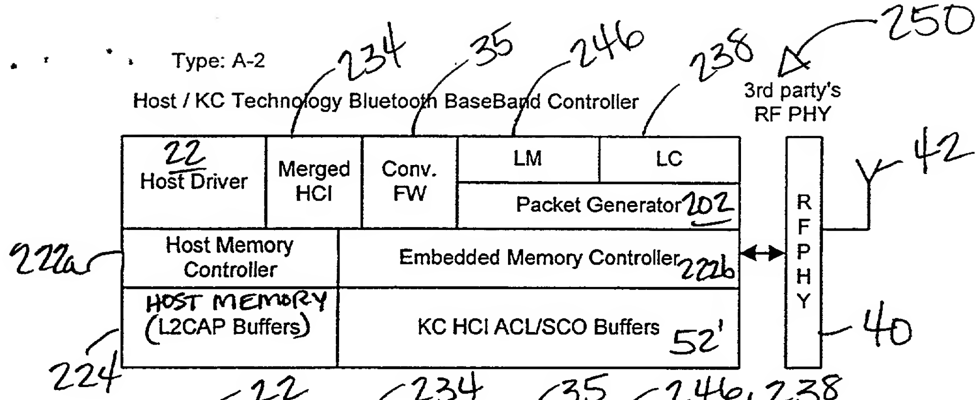


FIG. 15A

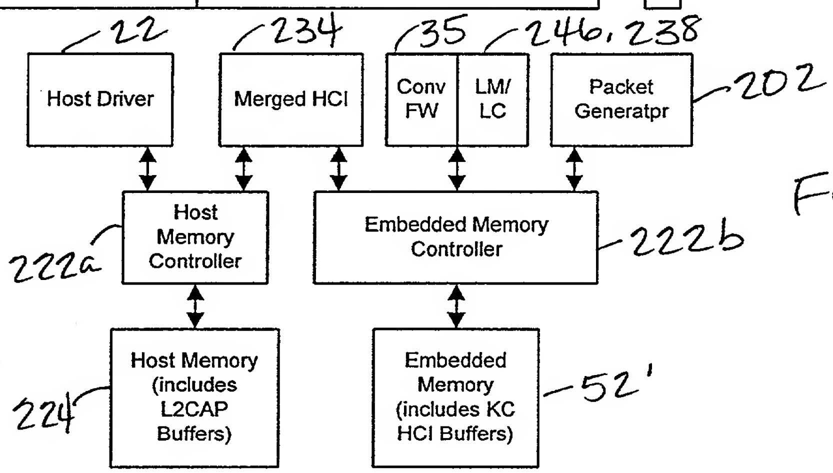


FIG. 15B.

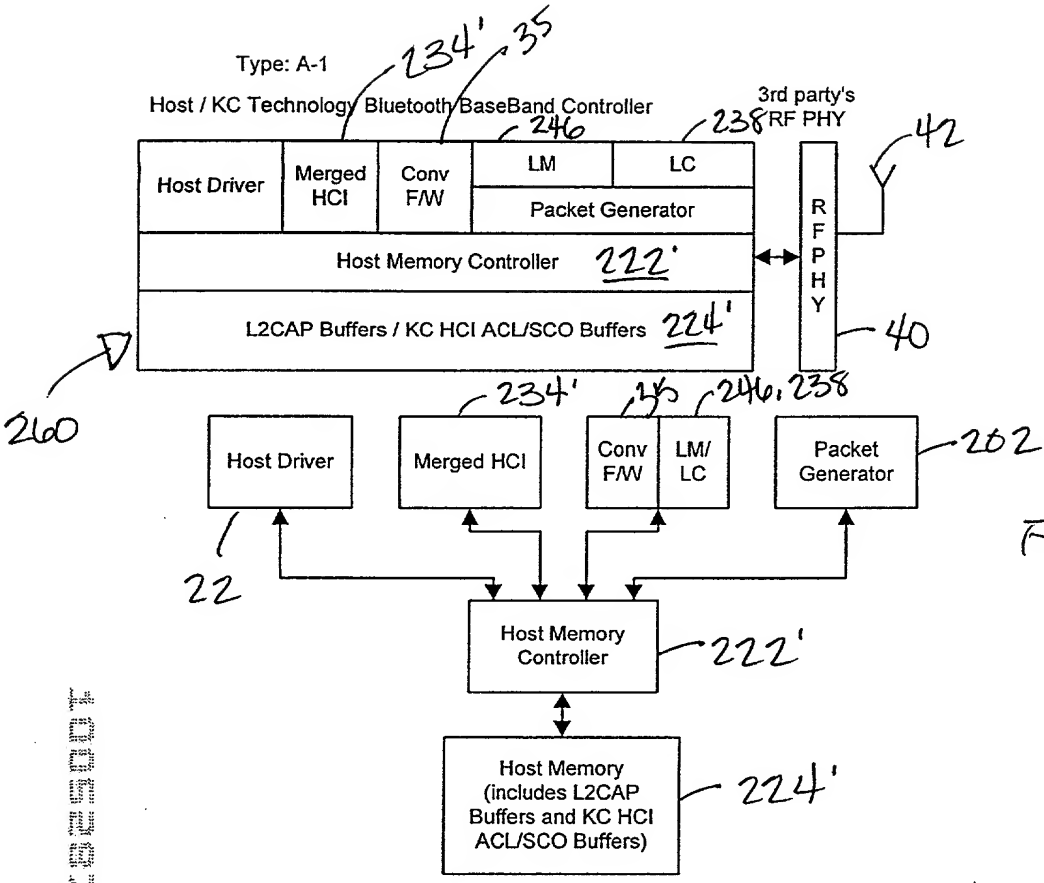


FIG. 16A

FIG. 16B

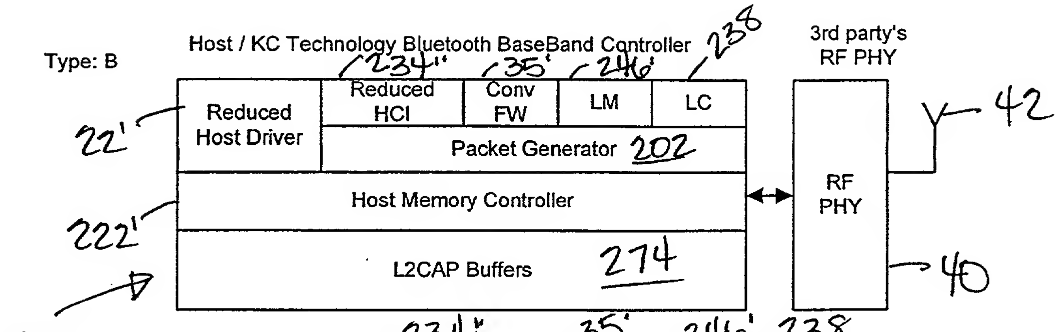


FIG. 17A

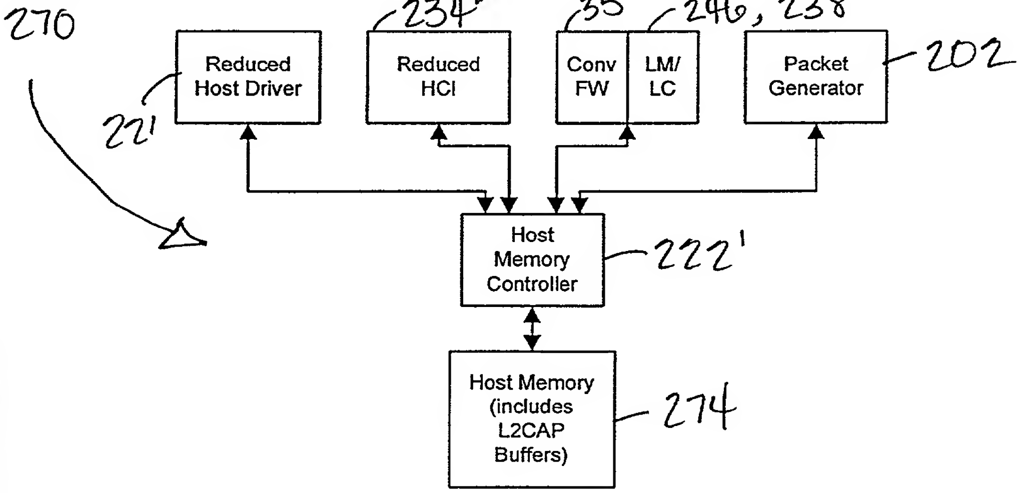


FIG. 17B.

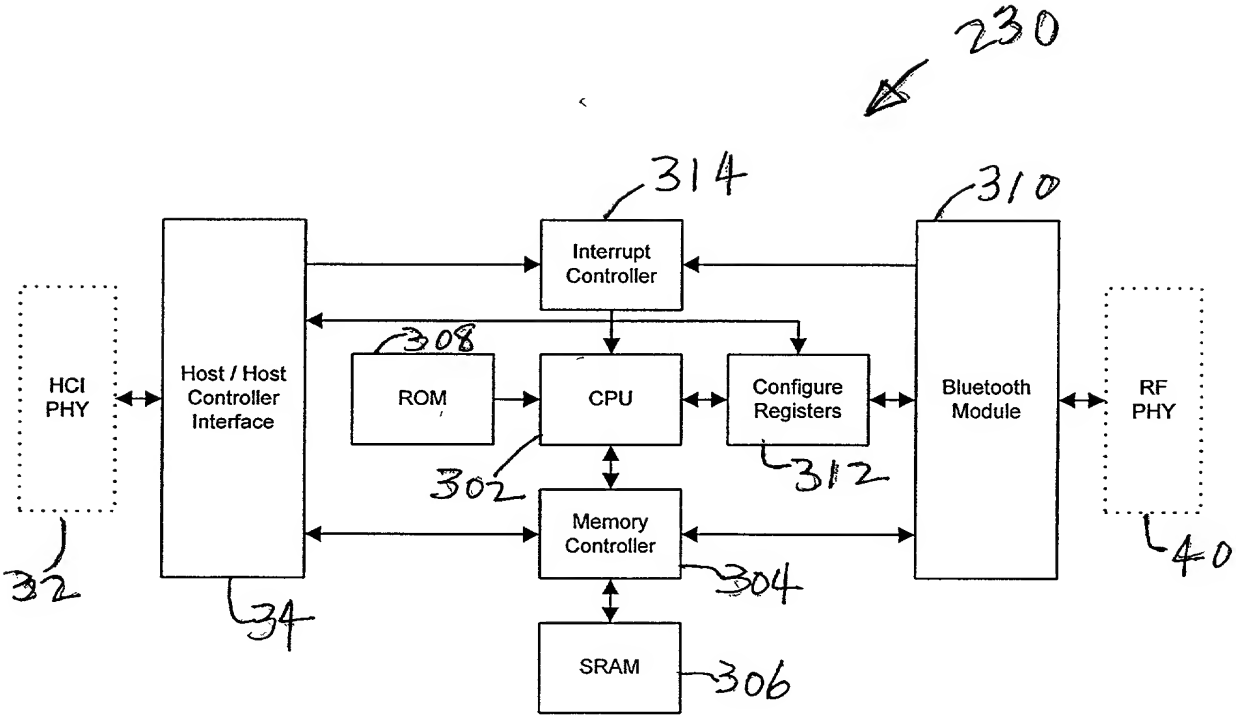
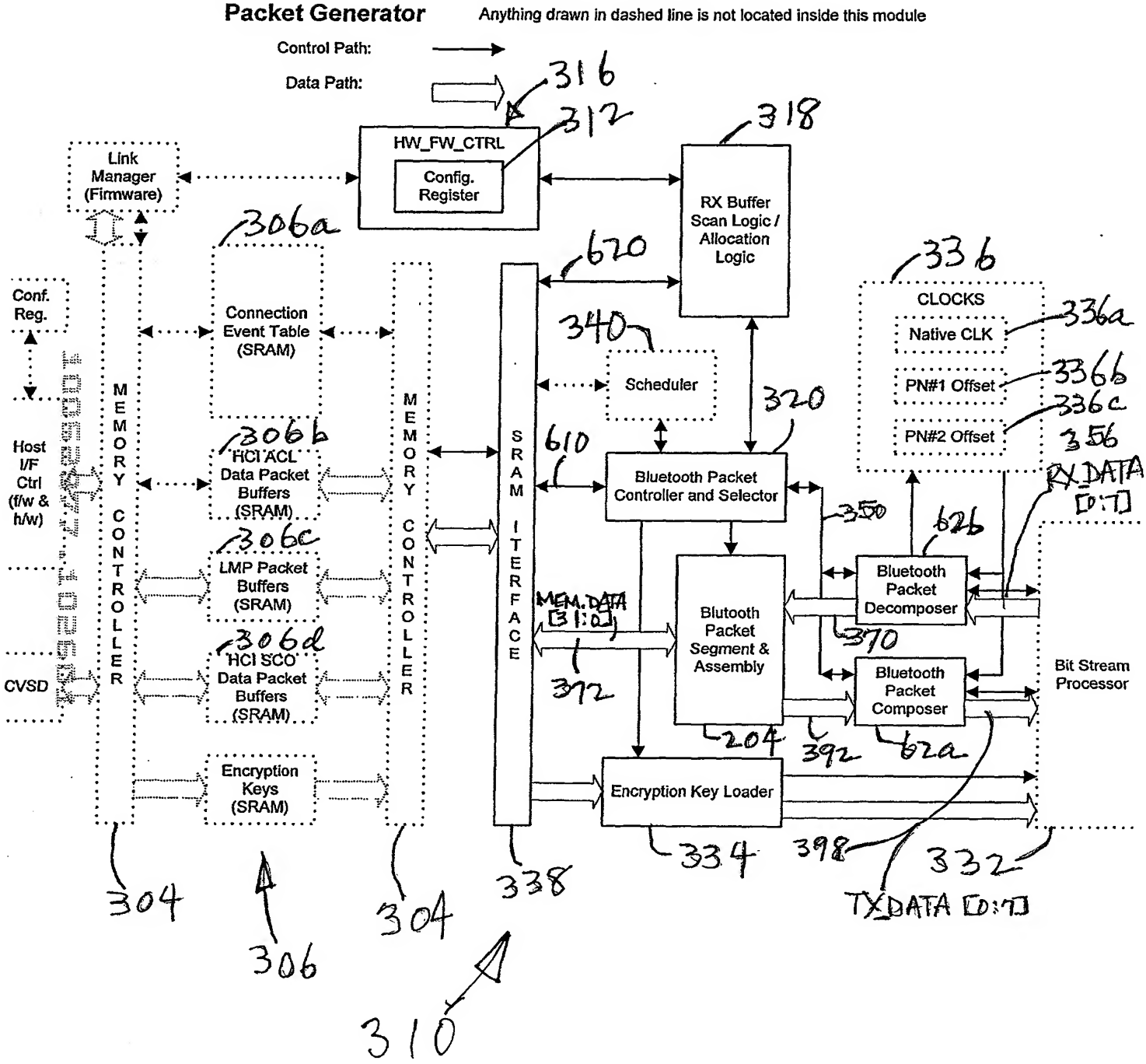

230
314
310
Interrupt Controller
HCI PHY
Host / Host Controller Interface
308
ROM
CPU
Configure Registers
Bluetooth Module
RF PHY
302
312
Memory Controller
304
SRAM
306
32
34
40

FIG. 18

FIG. 19


Packet Generator
Anything drawn in dashed line is not located inside this module
Control Path:
Data Path:
316
312
318
HW_FW_CTRL
Config. Register
Link Manager (Firmware)
RX Buffer Scan Logic / Allocation Logic
306a
620
336
CLOCKS
Native CLK
336a
PN#1 Offset
336b
PN#2 Offset
336c
356
Conf. Reg.
Connection Event Table (SRAM)
340
Scheduler
320
Host I/F Ctrl (f/w & h/w)
306b
HCI ACL Data Packet Buffers (SRAM)
610
Bluetooth Packet Controller and Selector
350
626
RX_DATA [0:7]
306c
LMP Packet Buffers (SRAM)
MEMORY CONTROLLER
SRAM INTERFACE
MEMORY CONTROLLER
MEM_DATA [31:0]
Blutooth Packet Segment & Assembly
Bluetooth Packet Decomposer
370
306d
HCI SCO Data Packet Buffers (SRAM)
CVSD
372
Bluetooth Packet Composer
Bit Stream Processor
204
392
626
Encryption Keys (SRAM)
Encryption Key Loader
334
398
332
TX_DATA [0:7]
304
304
338
306
310

FHS packet transmission: FIG. 23

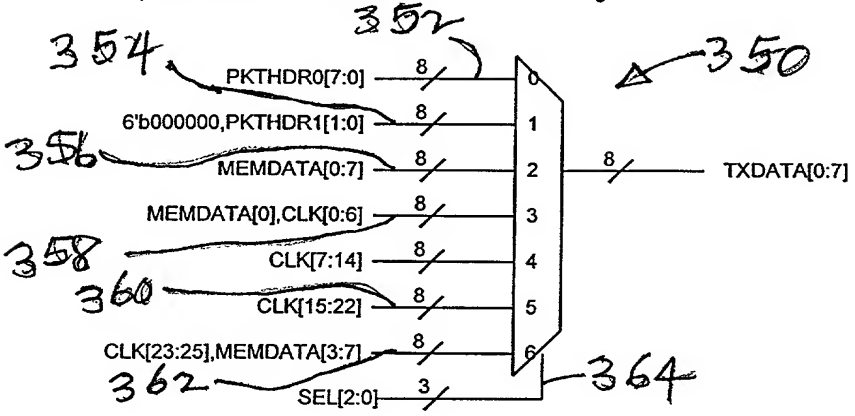


FIG. 24B

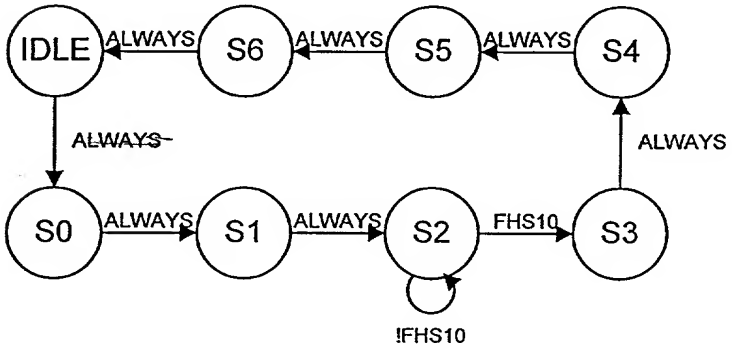


FIG. 24A

State	SEL[2:0]
IDLE	0
S0	0
S1	1
S2	2
S3	3
S4	4
S5	5
S6	6

FIG. 21

Incoming FHS Packet storage format:

31 … 24	23 … 16	15 … 8	7 … 0
na	0 1 1 1 0 BC 0 0 1		na
UAP [1:0]	SP[1:0] SR[1:0] na	LAP	
CLASS[9:0]	NAP[15:0]		UAP[7:2]
CLK[16:2]	AM_ADR [2:0]	CLASS[23:10]	
na	CLKOFFSET[16:2]	na PSM[2:0]	CLK[27:17]

Outgoing FHS Packet storage format:

31 … 24	23 … 16	15 … 8	7 … 0
na	0 1 1 1 0 BC 0 0		na
UAP [1:0]	SP[1:0] SR[1:0] na	LAP	
CLASS[9:0]	NAP[15:0]		UAP[7:2]
na	AM_ADR [2:0]	CLASS[23:10]	
na		PSM[2:0]	na

FIG. 20

Data Byte Sending Sequence in FHS packet:

Byte# & name	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
1:Packet Header 0	amadr0	amadr2	amadr2	pkttp0	pkttp1	pkttp2	pkttp3	flow
2:Packet Header 1	arqn	seqn	-	-	-	-	-	-
3:FHS 0	lap0	lap1	lap2	lap3	lap4	lap5	lap6	lap7
4:FHS 1	lap8	lap9	lap10	lap11	lap12	lap13	lap14	lap15
5:FHS 2	lap16	lap17	lap18	lap19	lap20	lap21	lap22	lap23
6:FHS 3	tbd0	tbd1	sr0	sr1	sp0	sp1	uap0	uap1
7:FHS 4	uap2	uap3	uap4	uap5	uap6	uap7	nap0	nap1
8:FHS 5	nap2	nap3	nap4	nap5	nap6	nap7	nap8	nap9
9:FHS 6	nap10	nap11	nap12	nap13	nap14	nap15	clss0	clss1
10:FHS 7	clss2	clss3	clss4	clss5	clss6	clss7	clss8	clss9
11:FHS 8	clss10	clss11	clss12	clss13	clss14	clss15	clss16	clss17
12:FHS 9	clss18	clss19	clss20	clss21	clss22	clss23	amad0	amad1
13:FHS 10	amad2	clk0	clk1	clk2	clk3	clk4	clk5	clk6
14:FHS 11	clk7	clk8	clk9	clk10	clk11	clk12	clk13	clk14
15:FHS 12	clk15	clk16	clk17	clk18	clk19	clk20	clk21	clk22
16:FHS 13	clk23	clk24	clk25	pgscn0	pgscn1	pgscn2	-	-

FIG. 22

DM1 packet ransmission:

TXDATA[7:0], RXDAT, [7:0] : please refer to the definition on page 2 of FHS packet transmission.
Packet Generator, Bit · t eam Process Module diagram:please refer to the diagram on page 2 of FHS packet transmission.

Data Byte Sending Sequ nce in DM1 packet: FIG. 25

Byte# & name	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
1:Packet Header 0	amadr0	amadr2	amadr1	pkttp0	pkttp1	pkttp2	pkttp3	flow
2:Packet Header 1	arqn	seqn	-	-	-	-	-	-
3:Payload Header 0	l_ch 0	l_ch 1	pld_flow	length0	length1	length2	length3	length4
4:ACL data 1	data bit 0	data bit 1	data bit 2	data bit 3	data bit 4	data bit 5	data bit 6	data bit 7
5: ACL data 2	data bit 0	data bit 1	data bit 2	data bit 3	data bit 4	data bit 5	data bit 6	data bit 7
...	...	...	...	...	...	...	...	...
n+3:ACL data n	data bit 0	data bit 1	data bit 2	data bit 3	data bit 4	data bit 5	data bit 6	data bit 7

n; data length

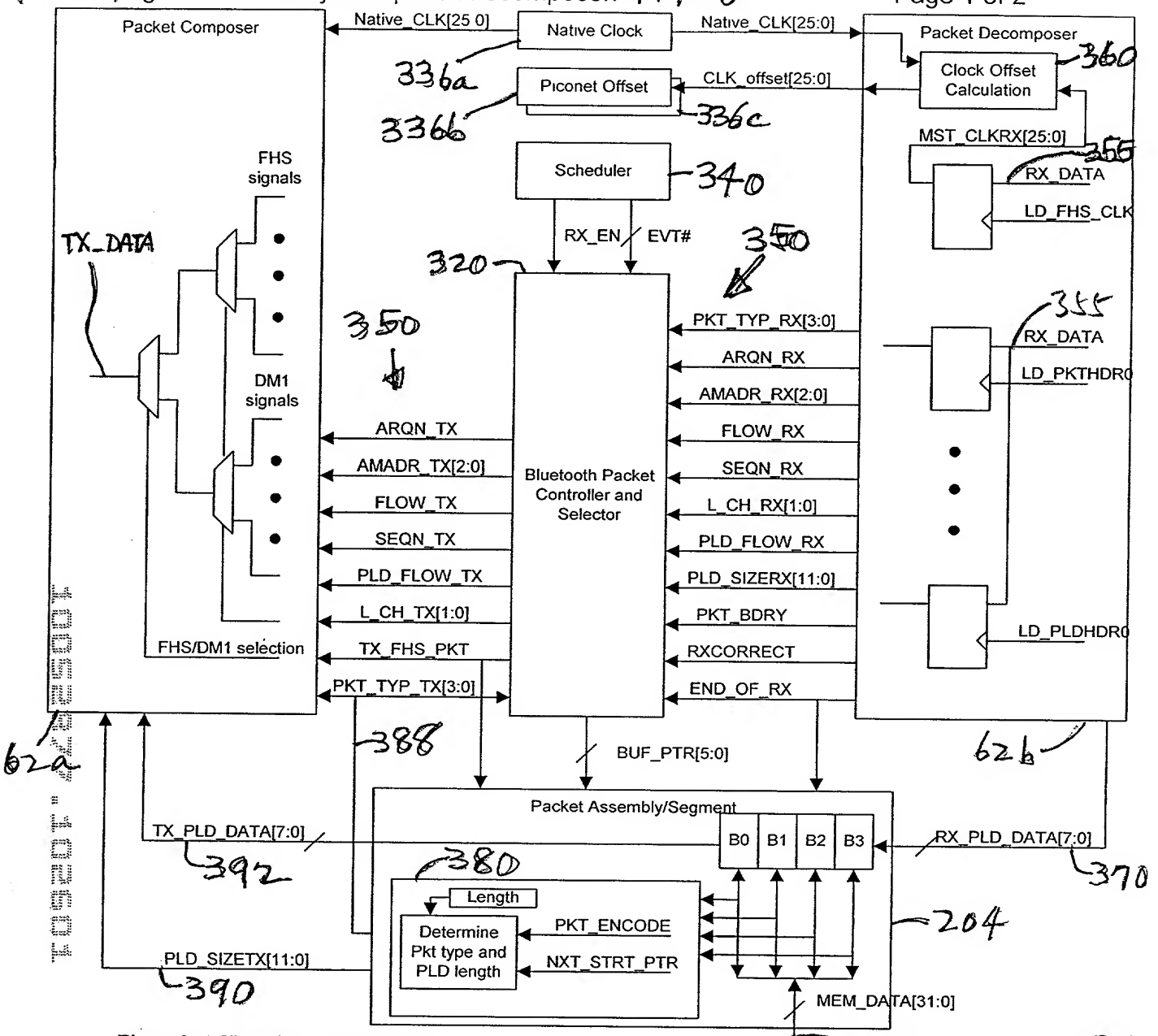
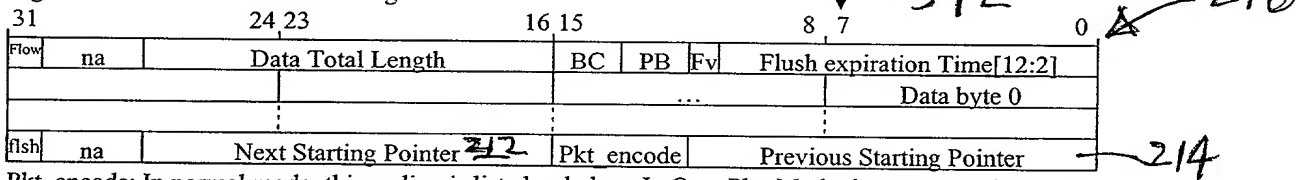


Figure 3: ACL Buffer while Sending

FIG. 9A



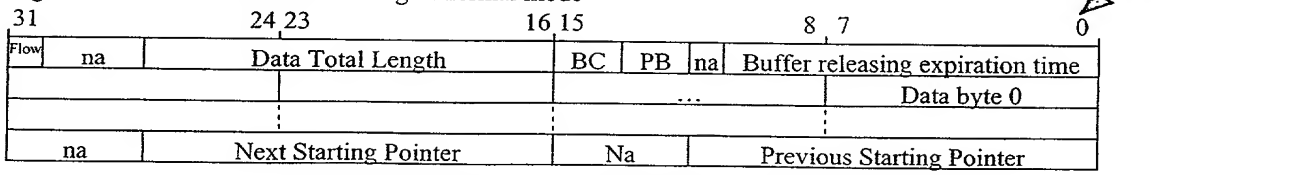
Pkt_encode: In normal mode, this coding is listed as below. In One_Pkt_Mode, it represents the packet type code that is defined in the Bluetooth specification.

Pkt_encode[3:2]: 00 – AUX1 only; 01 – DM only; 10 – DH only; 11 – Automatic best fit

Pkt_encode[1:0]: 00 – Single slot packet; 01 – 3-slot packet; 10 – 5-slot packet; 11 – Reserved

Figure 5: ACL Buffer after Receiving in normal mode

FIG. 9B



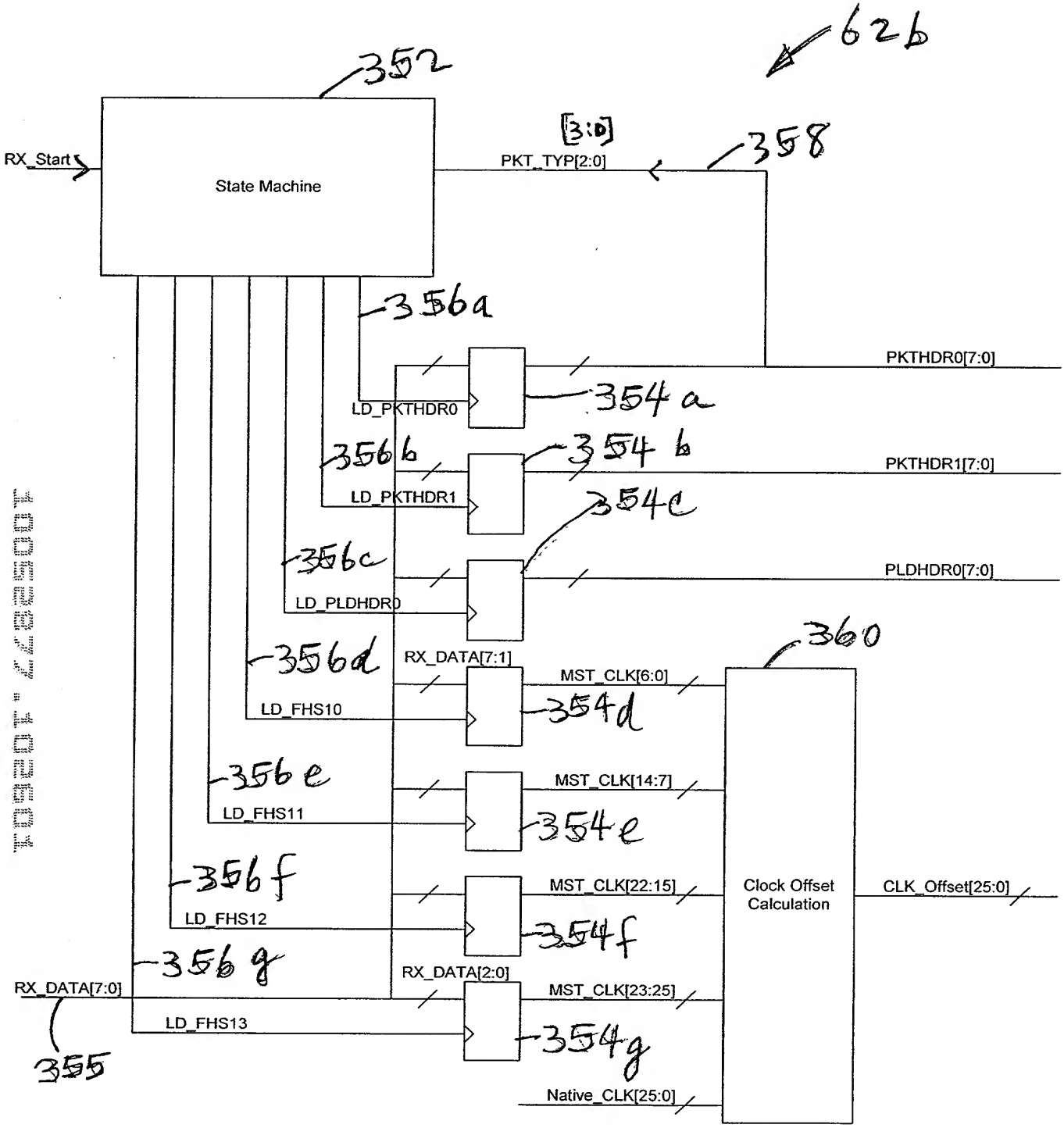

352
362b
[3:0]
358
RX_Start
State Machine
PKT_TYP[2:0]
356a
PKTHDR0[7:0]
LD_PKTHDR0
354a
356b
354b
PKTHDR1[7:0]
LD_PKTHDR1
354c
356c
PLDHDR0[7:0]
LD_PLDHDR0
360
356d
RX_DATA[7:1]
MST_CLK[6:0]
LD_FHS10
354d
356e
MST_CLK[14:7]
LD_FHS11
354e
356f
MST_CLK[22:15]
Clock Offset Calculation
CLK_Offset[25:0]
LD_FHS12
354f
356g
RX_DATA[2:0]
RX_DATA[7:0]
MST_CLK[23:25]
LD_FHS13
354g
355
Native_CLK[25:0]

FIG. 27

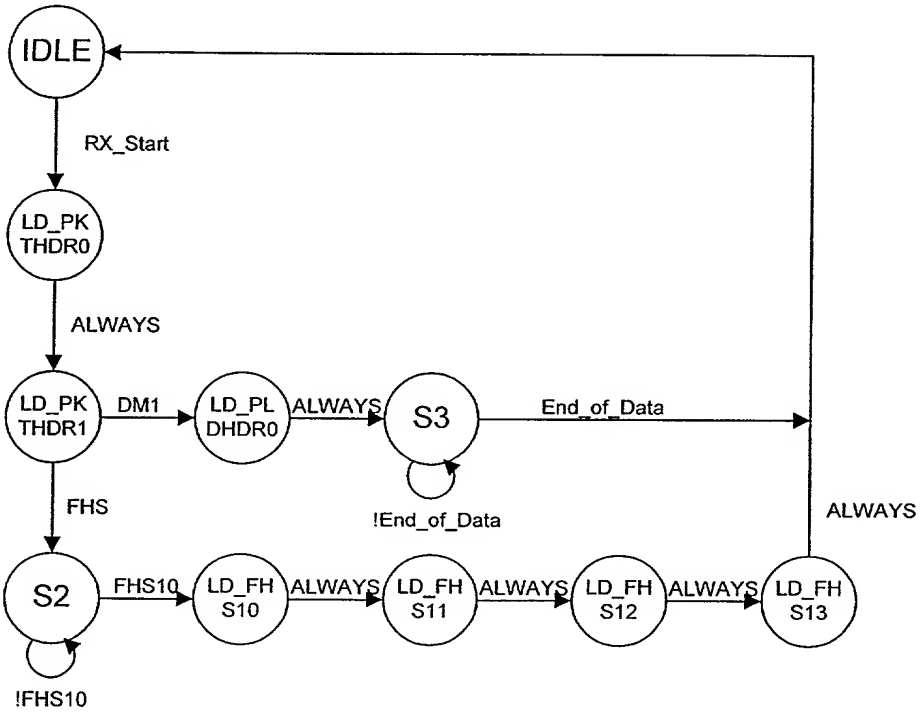


Fig. 28

FIG. 29A

Determine Pkt type and PLD length

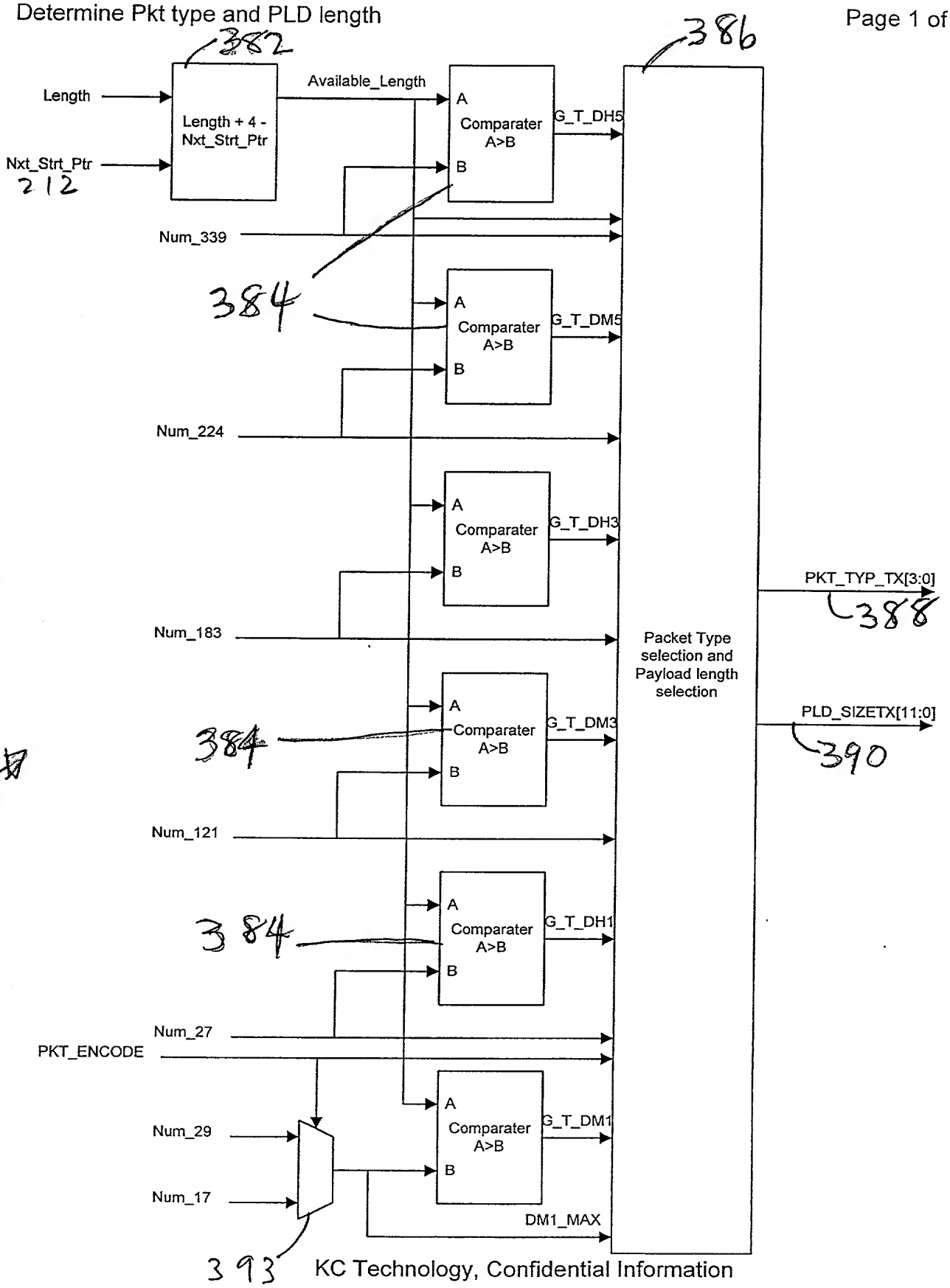
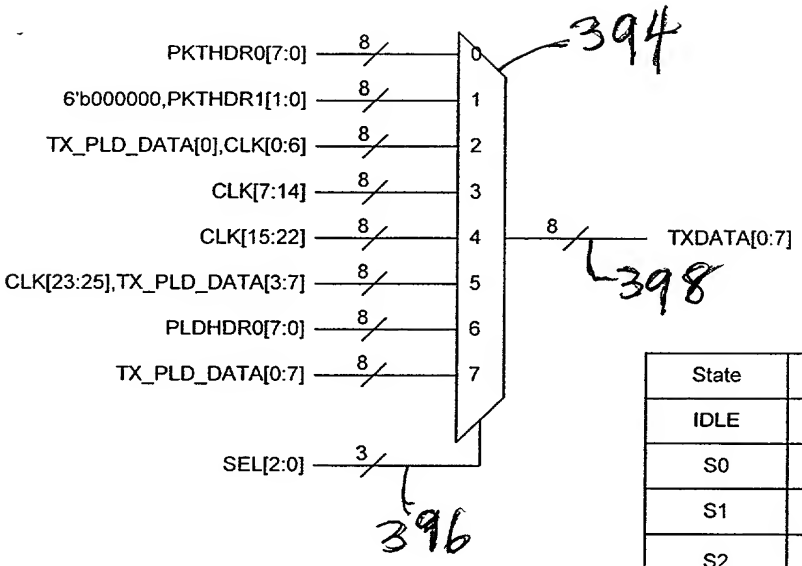


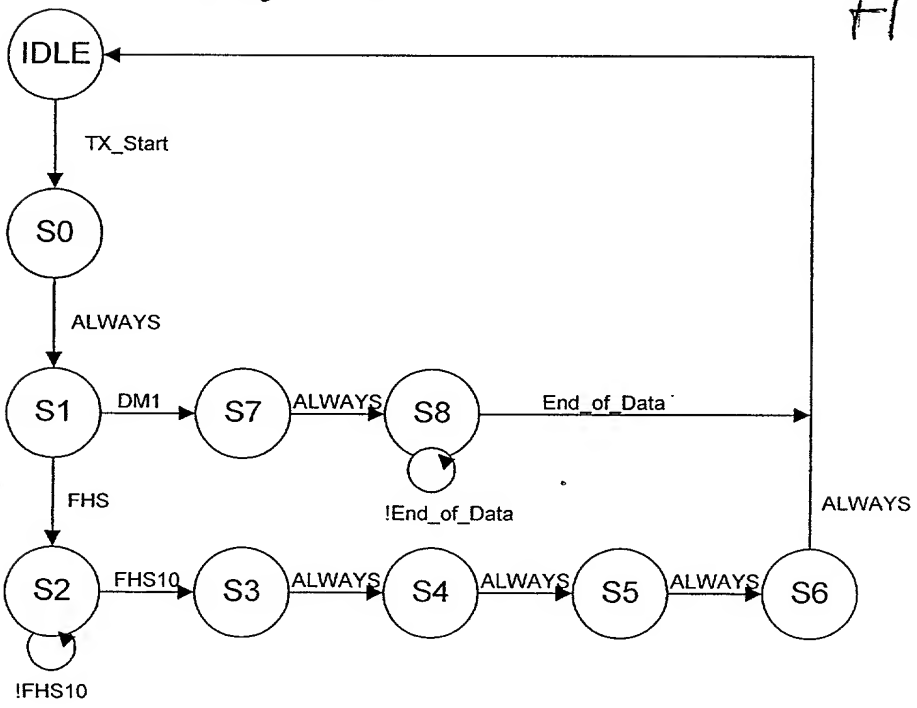
FIG.30



State	SEL[2:0]
IDLE	0
S0	0
S1	1
S2	7
S3	2
S4	3
S5	4
S6	5
S7	6
S8	7

FIG.31A

FIG.31B



Dual pointer buffer control scheme (1)

FIG. 32

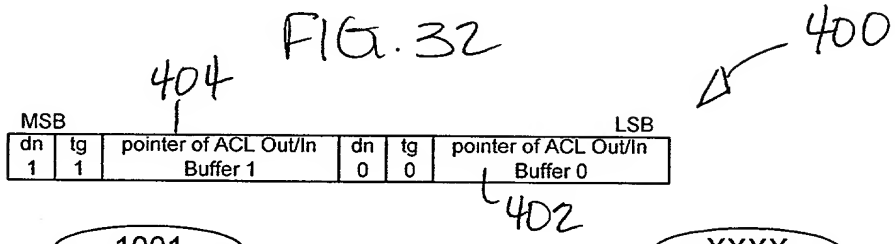


FIG. 35B

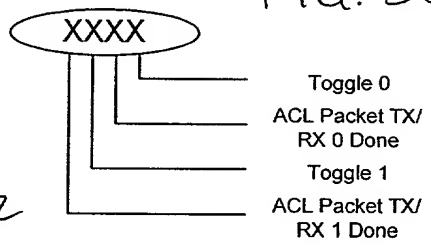


FIG. 35A

- "6'b111XXX" is defined as NULL pointer.
- When Done bit is 1, firmware can update the pointer byte and hardware can only read this byte.
- When Done bit is 0, hardware can update the pointer byte and firmware can only read this byte.
- Toggle bit is changed by the hardware, it is toggled everytime when hardware finishes a task.
- This control scheme applies to the LMP out buffers as well.

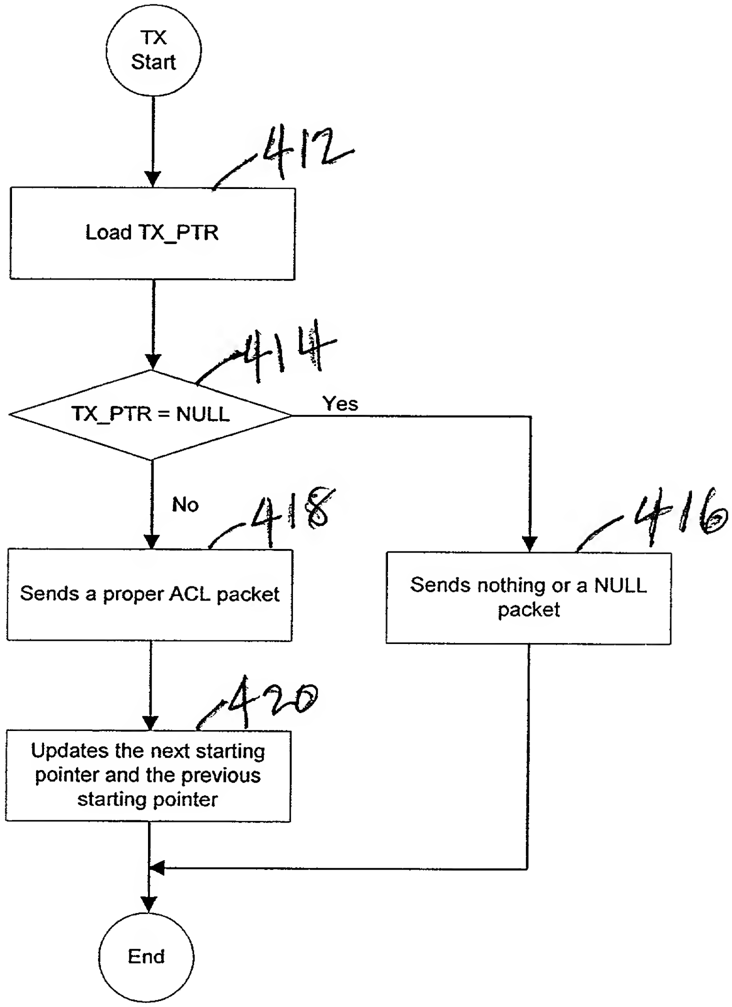
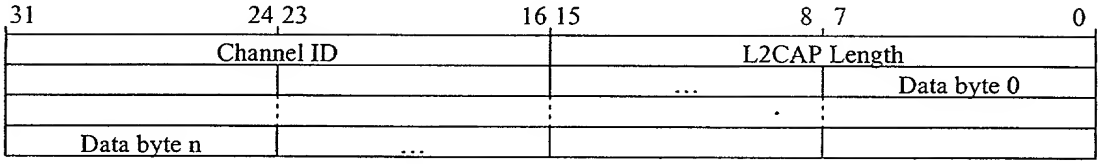


FIG. 33

FIG. 34A

L2CAP Packet Format



ACL Buffer while Sending

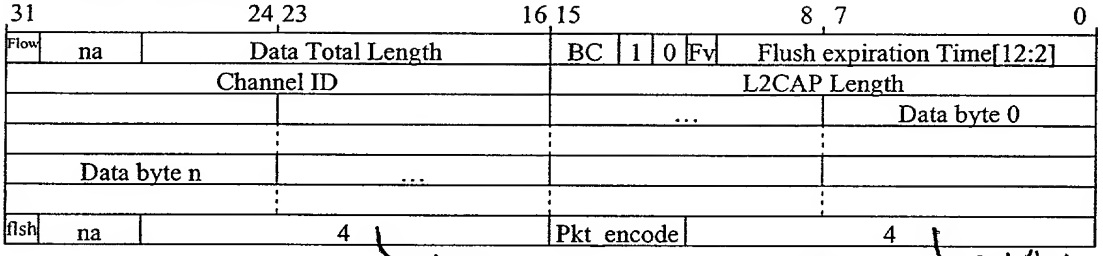
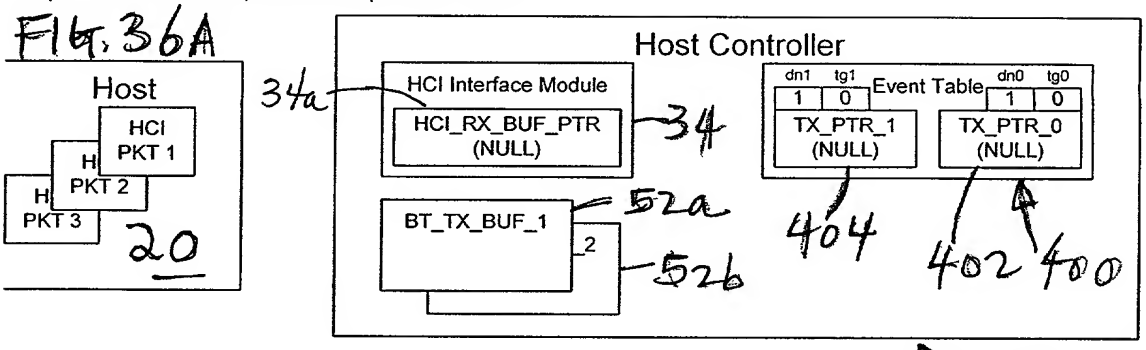


FIG. 34B

Step 1: Afterinitialization, the value of pointers is 'NULL"

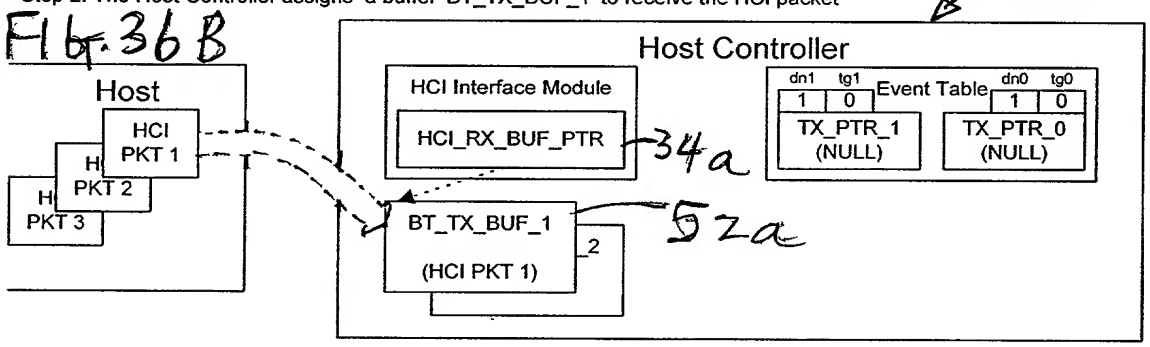
Fig. 36A



230

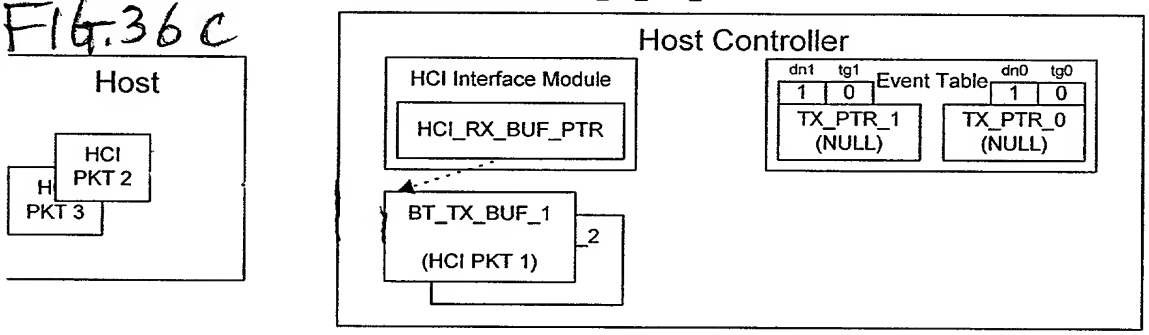
Step 2: The Host Controller assigns a buffer 'BT_TX_BUF_1' to receive the HCI packet

Fig. 36B



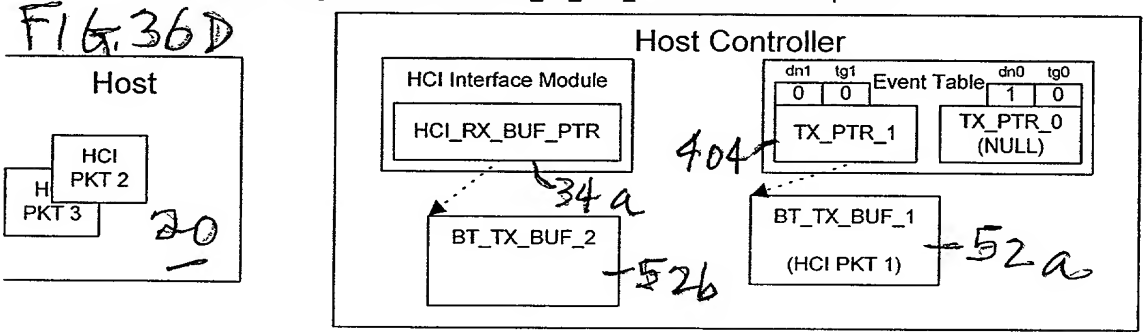
Step 3: The HCI packet 'HCI PKT 1' is stored in the buffer 'BT_TX_BUF_1'

Fig. 36C



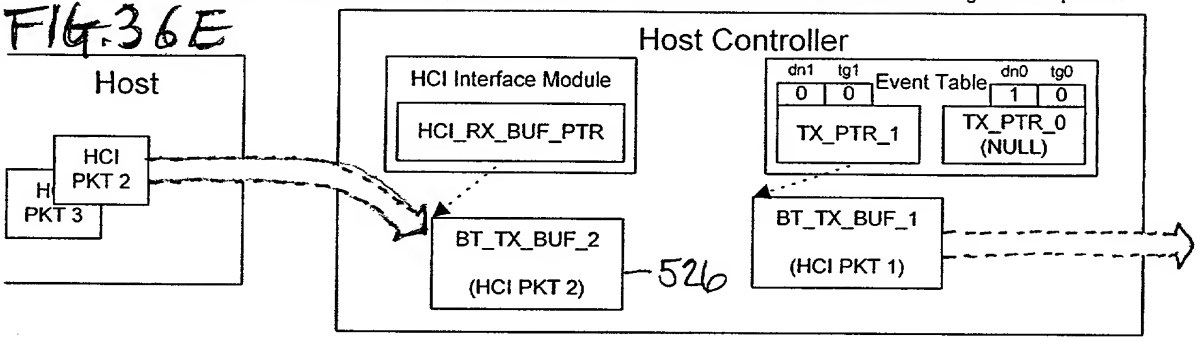
Step 4: The Host Controller assigns another buffer 'BT_TX_BUF_2' to receive the HCI packet

Fig. 36D



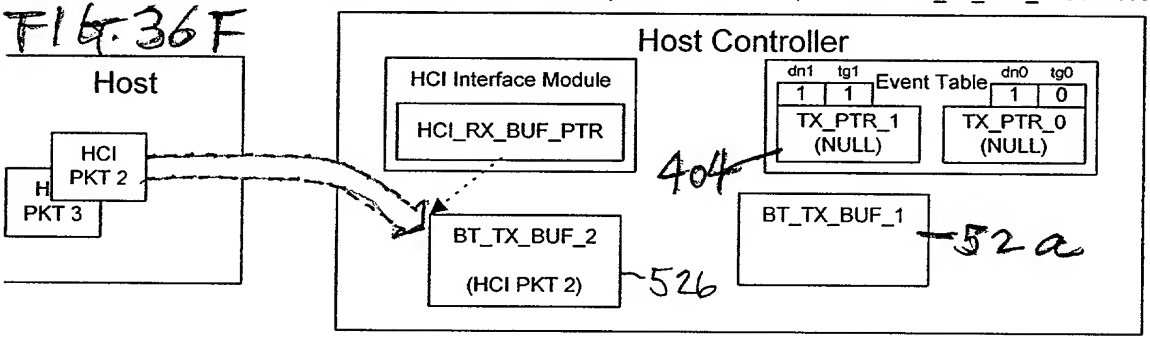
Step 5: While the Bluetooth Module is sending the HCI packet 1, the HCI Interface Module is receiving the HCI packet 2.

FIG.36E



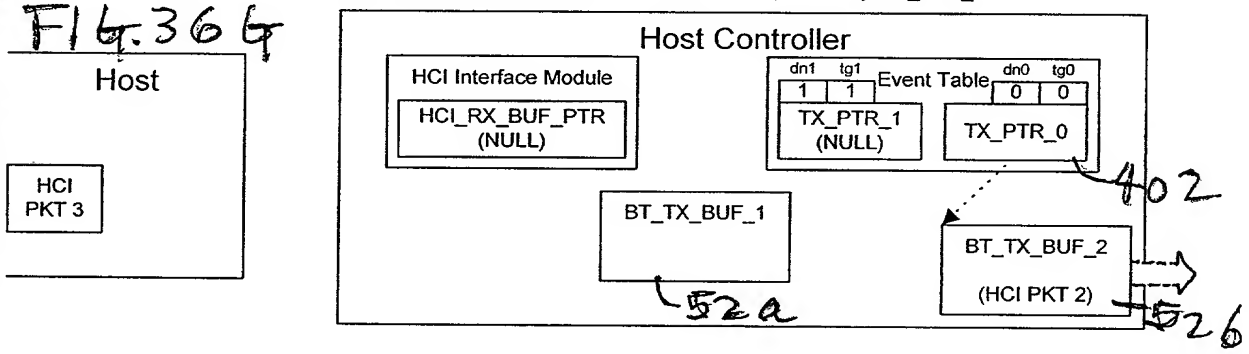
Step 6: Assuming that the HCI packet 1 is sent before the HCI packet 2 is received, the buffer 'BT_TX_BUF_1' is released.

FIG.36F



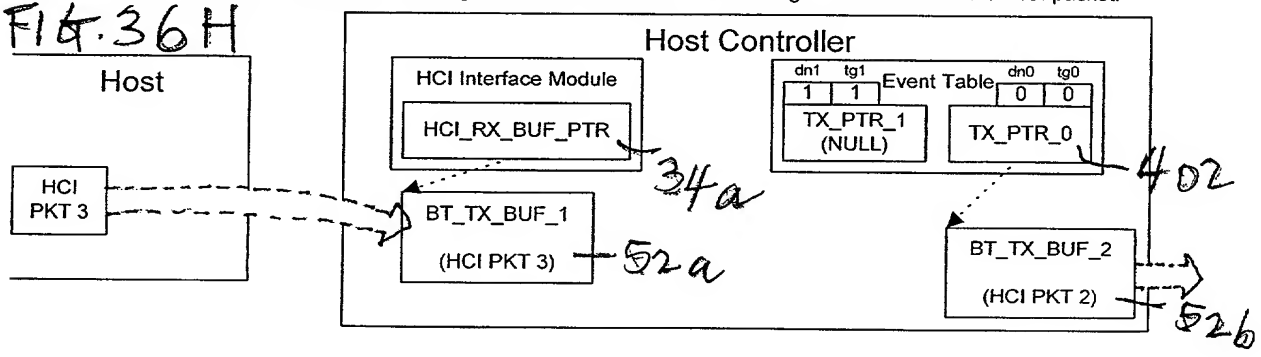
Step 7: After the HCI packet 2 has been received, the buffer 'BT_TX_BUF_2' is pointed by 'TX_PTR_0'.

FIG.36G



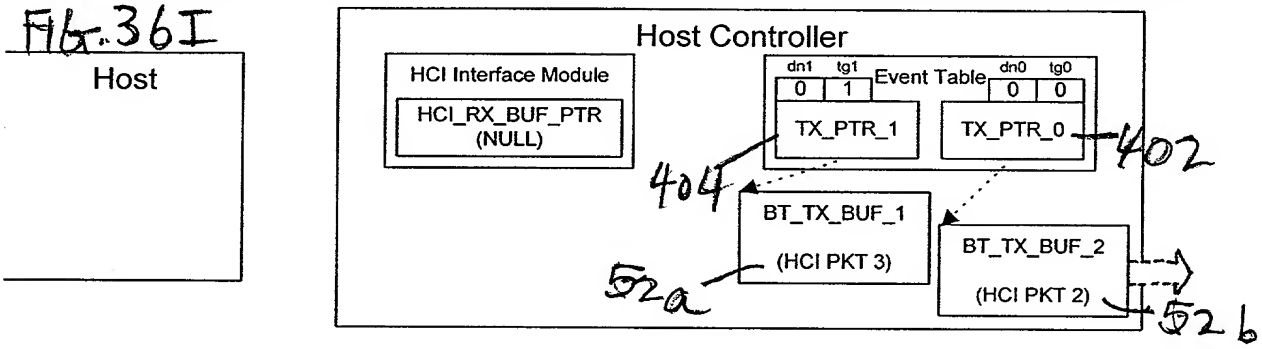
Step 8: The free buffer 'BT_TX_BUF_1' is assigned to the HCI Interface Module again to receive another HCI packet.

FIG.36H



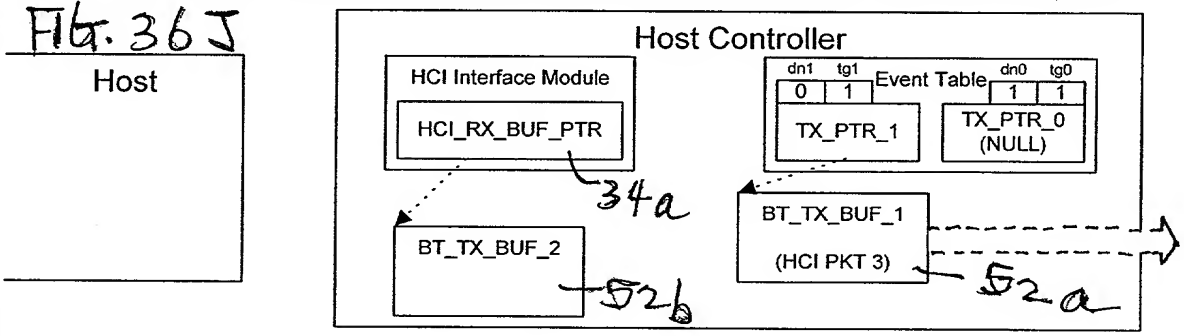
Step 9: Assuming that the HCI packet 3 is received before the HCI packet 2 is sent, 'TX_PTR_1' points to buffer 'BT_TX_BUF_1'.

FIG. 36I



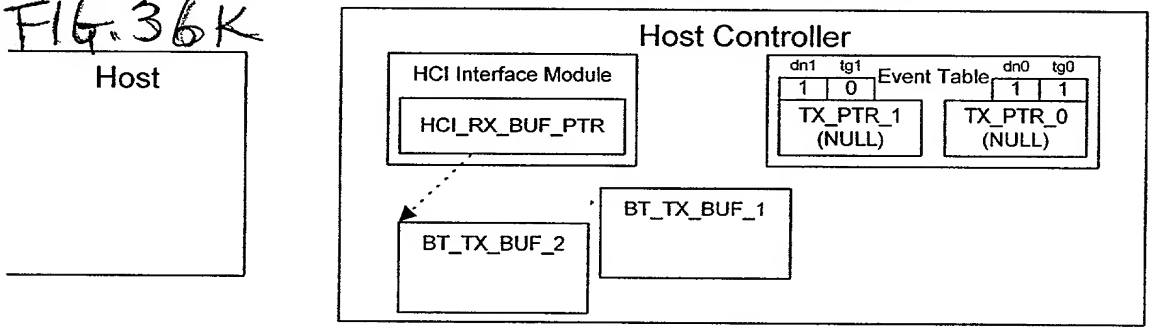
Step 10: The free buffer 'BT_TX_BUF_2' is assigned to the HCI Interface Module again to receive another HCI packet.

FIG. 36J



Step 11: After the HCI packet 3 has been transmitted, the buffer 'BT_TX_BUF_1' will be released.

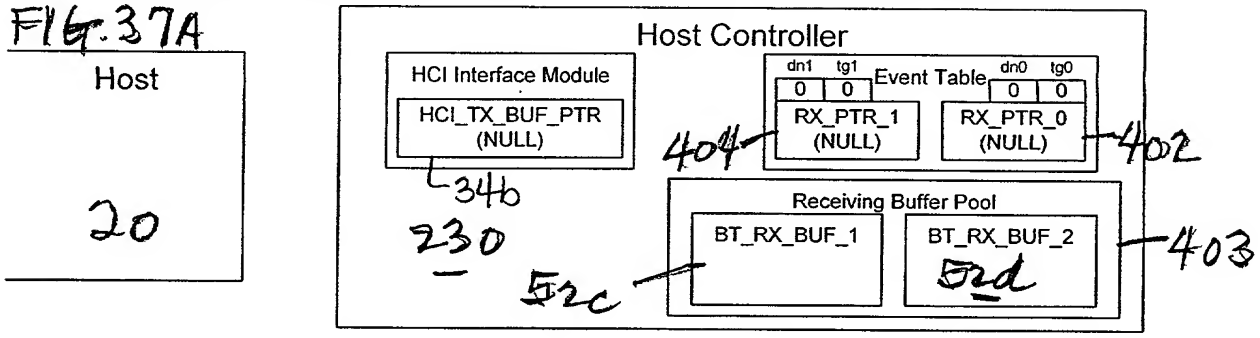
FIG. 36K



Now, the state of done and toggle is "1011". For the next transmisstion, 'TX_PTR_0' is selected to point the next outgoing packet.

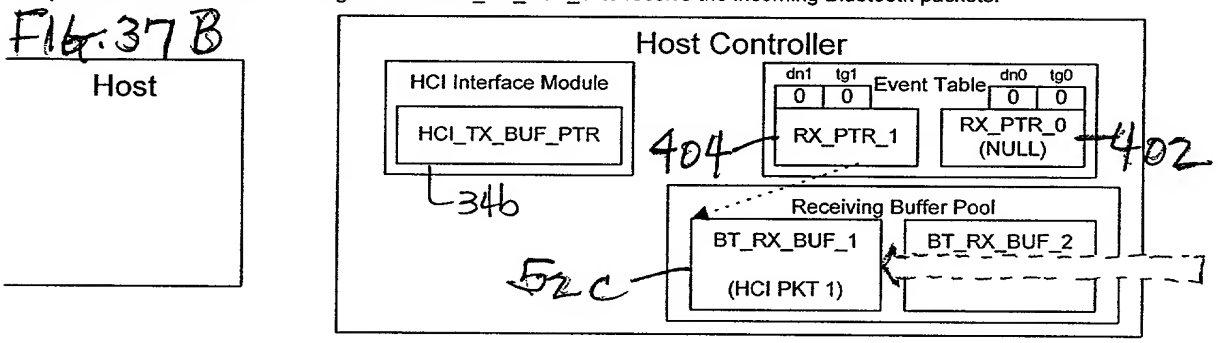
Step 1: After initialization, the value of pointers is 'NULL". Assuming that two receiving buffer are available.

FIG.37A



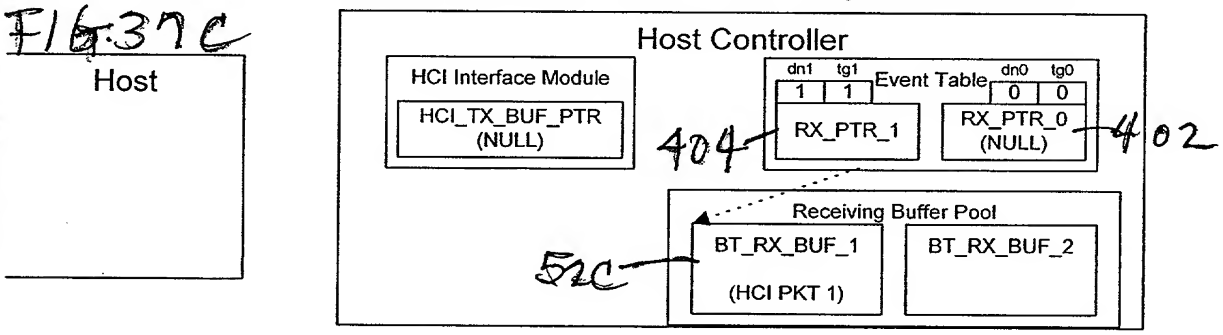
Step 2: The Bluettoth Module assigns buffer 'BT_RX_BUF_1' to receive the incoming Bluetooth packets.

FIG.37B



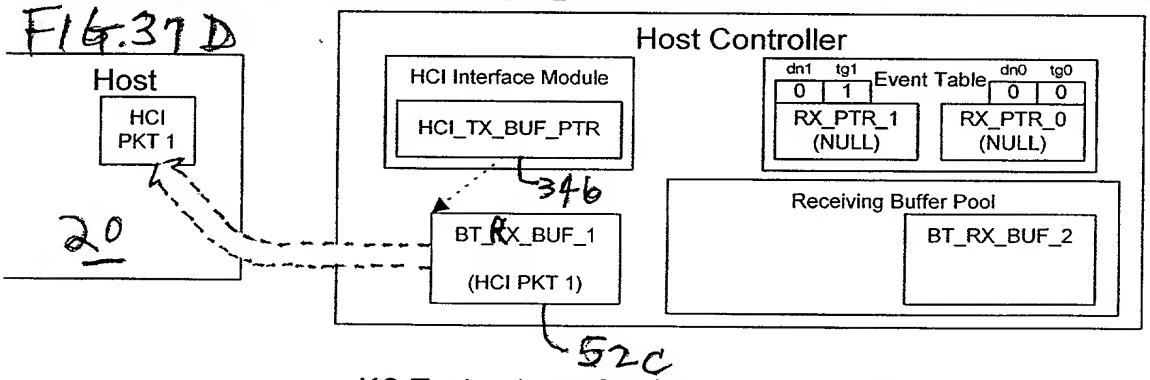
Step 3: The buffer 'BT_TX_BUF_1' is released when any one of the three buffer releasing conditions is detected.

FIG.37C



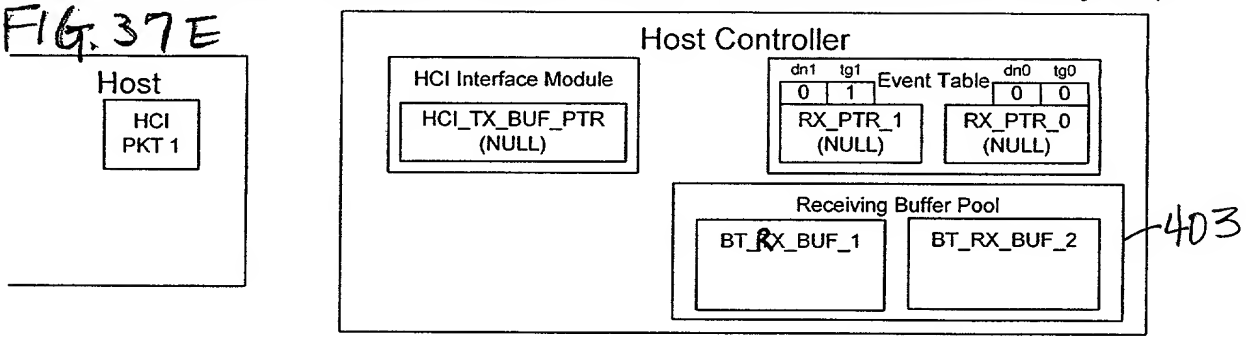
Step 4: Firmware releases this buffer buffer 'BT_TX_BUF_1' and sends it to the HCI Interface Module. Then sets the done bit to 0.

FIG.37D



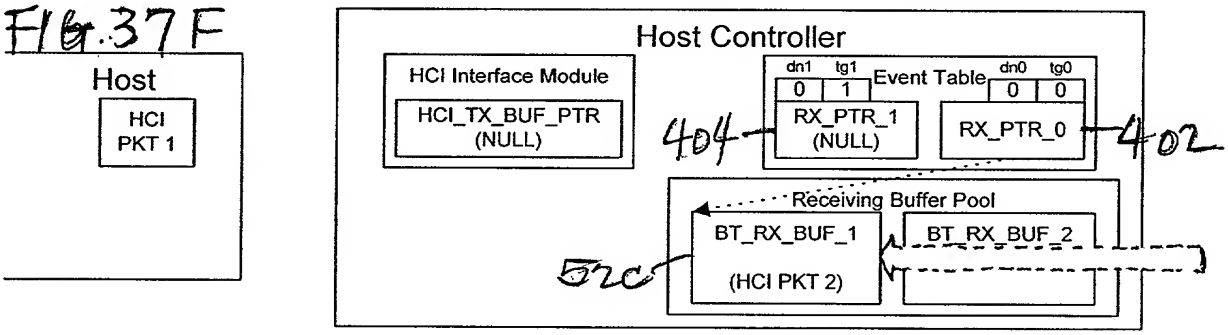
Step 5: After the HCI packet 1 is sent to the Host, buffer 'BT_RX_BUF_1' is released and put back to the receiving buffer pool

FIG.37E



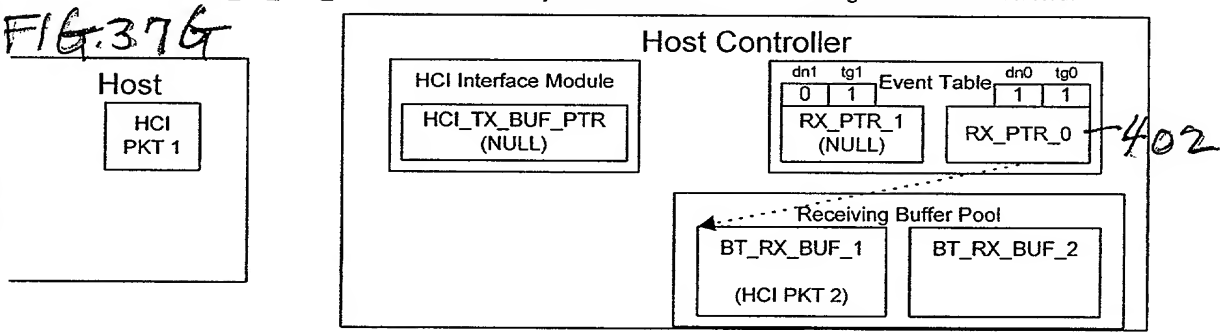
Step 6: The Bluettoth Module assigns buffer 'BT_RX_BUF_1' to receive the incoming Bluetooth packets.

FIG.37F



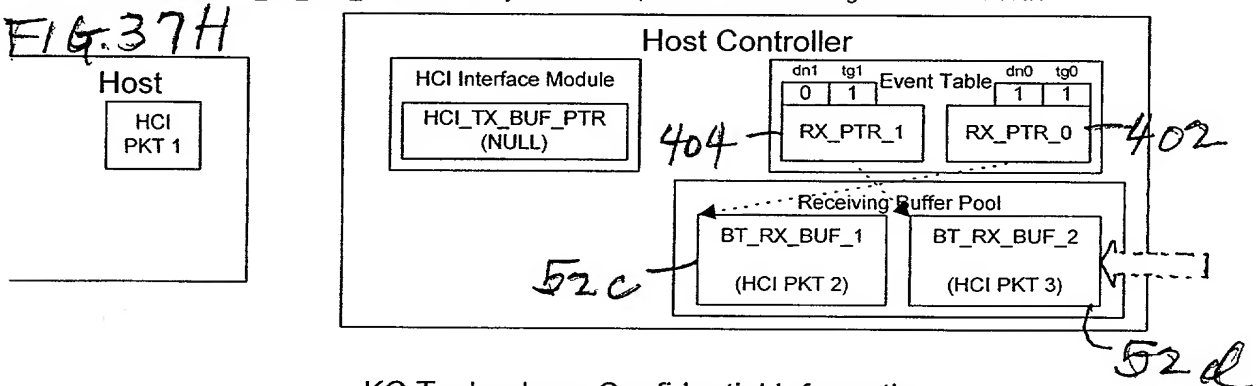
Step 7: The buffer 'BT_TX_BUF_1' is released when any one of the three buffer releasing conditions is detected.

FIG.37G



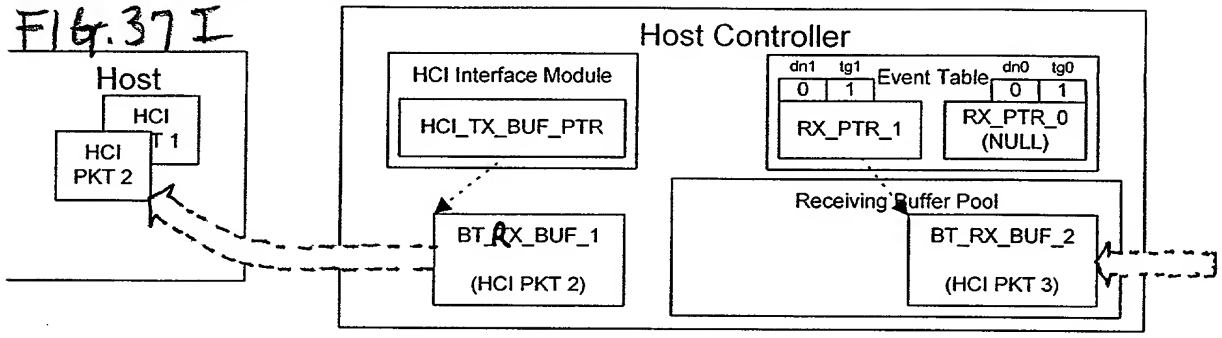
Step 8: Before buffer 'BT_RX_BUF_1' is removed by the firmware, another buffer is assigned to receive data.

FIG.37H



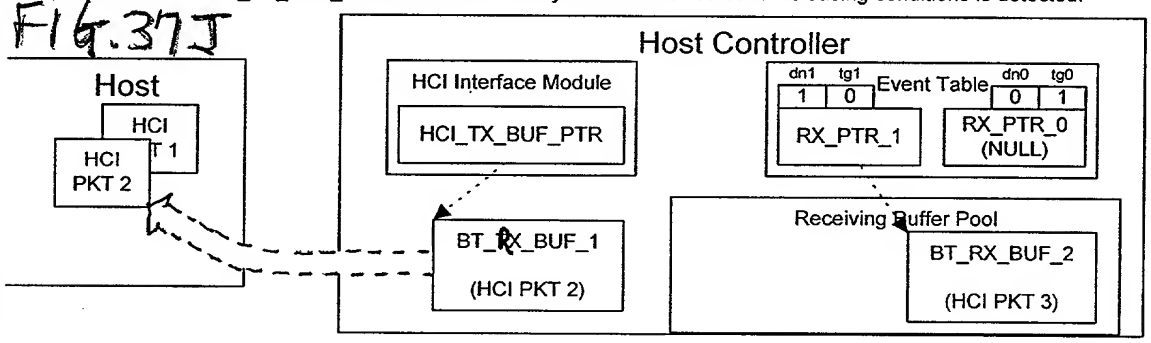
Step 9: Firmware releases this buffer 'BT_TX_BUF_1' and sends it to the HCI Interface Module Then sets the done bit to 0.

FIG.37I



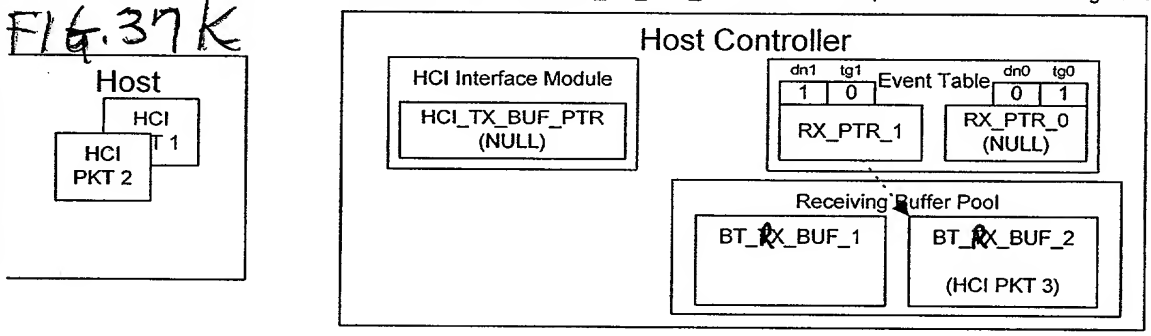
Step 10: The buffer 'BT_TX_BUF_2' is released when any one of the three buffer releasing conditions is detected.

FIG.37J



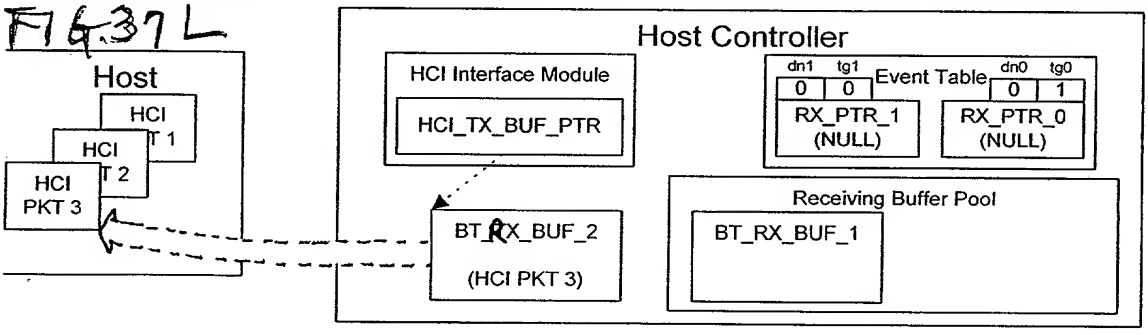
Step 11: After the HCI packet 1 is sent to the Host, buffer 'BT_RX_BUF_1' is released and put back to the receiving buffer pool.

FIG.37K



Step 12: Firmware releases this buffer buffer 'BT_TX_BUF_1' and sends it to the HCI Interface Module. Then sets the done bit to 0.

FIG.37L



Dual Pointer Buffer Scheme 1: Hardware implementation

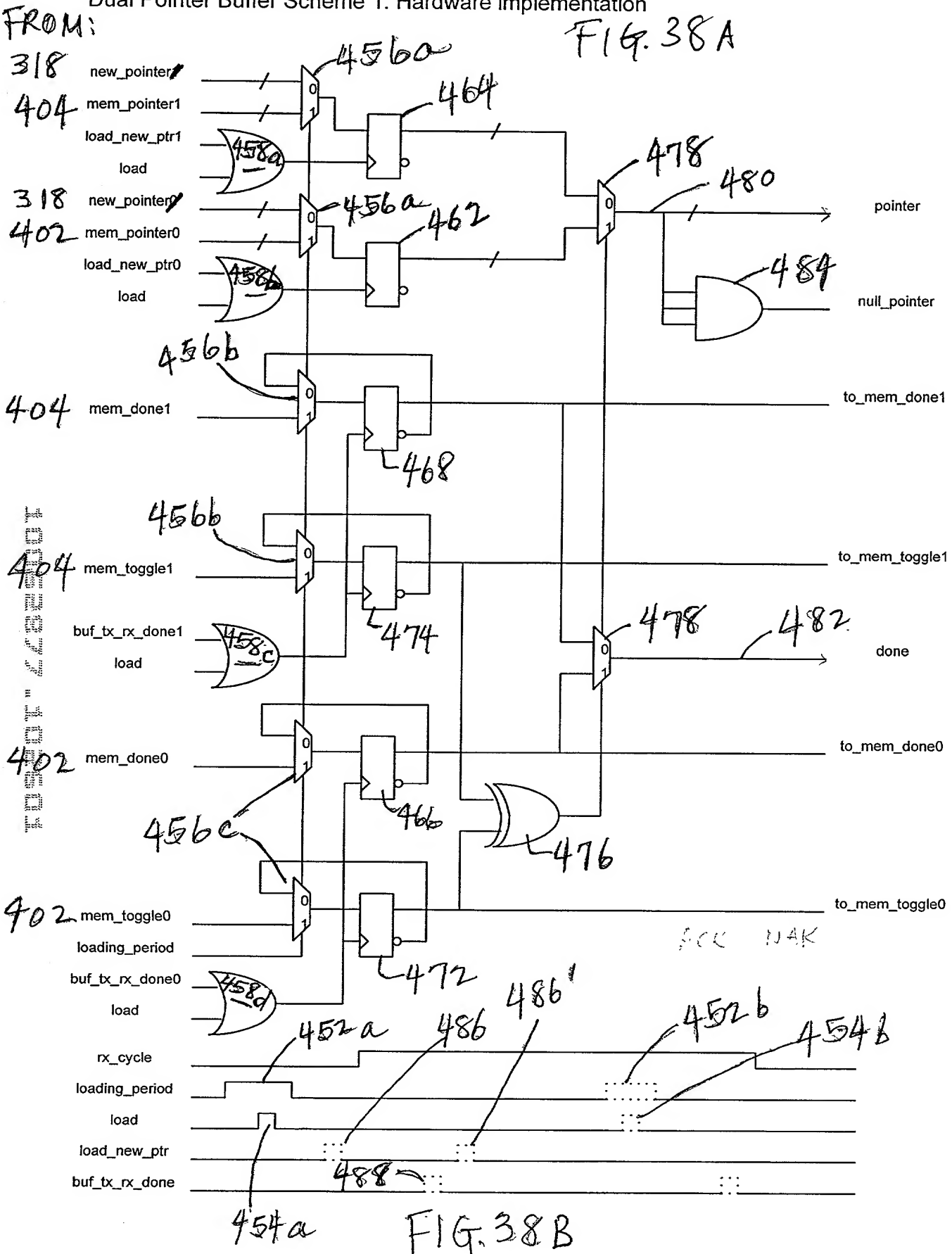


FIG. 38A

FIG. 38B

case 1: An interrupt which is generated by the Packet Controller of the BT module indicates that an incoming HCI packet is received

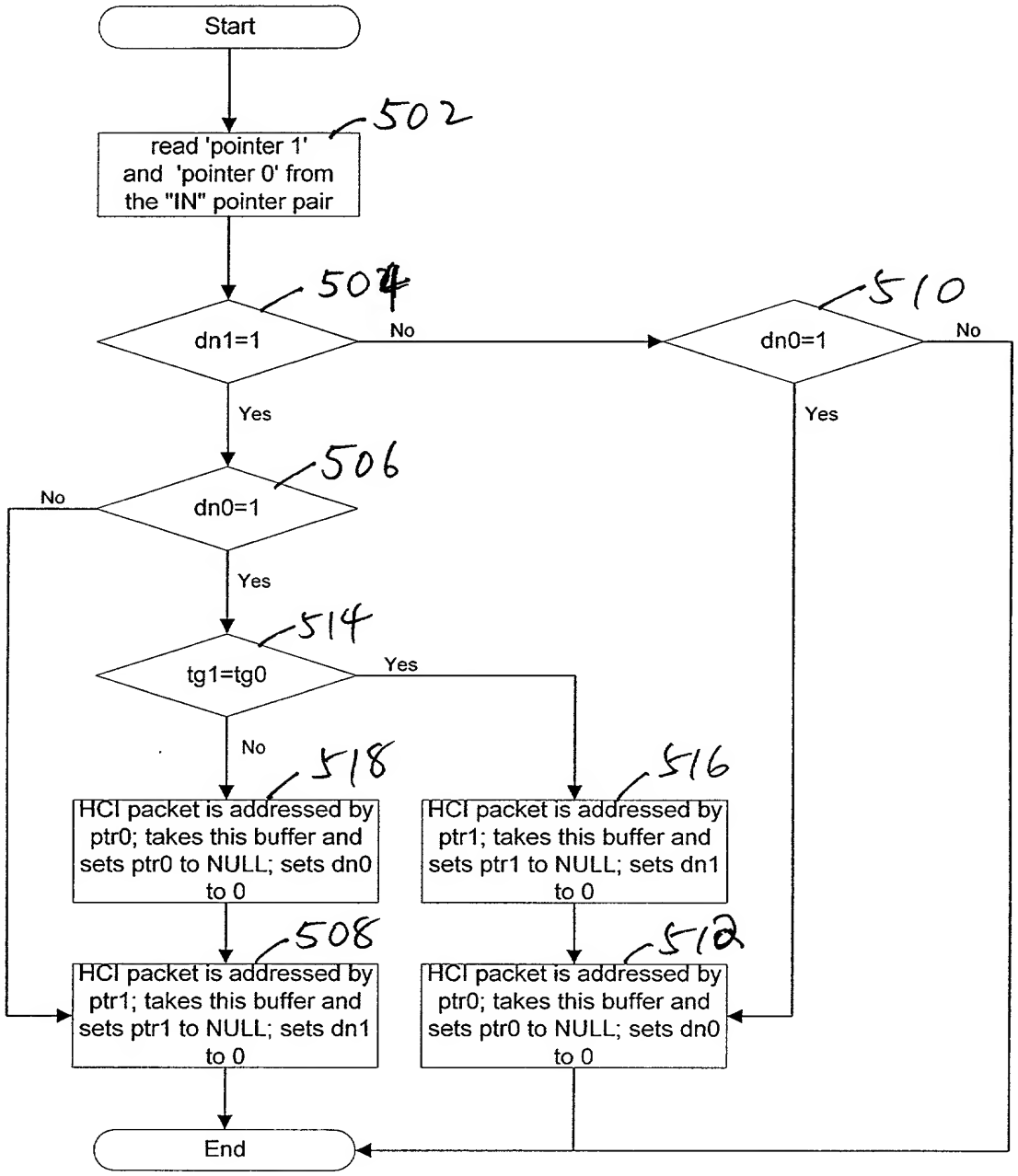


FIG. 39

case 3: An interrupt which is generated by the HCI module indicates that an outgoing HCI packet is received

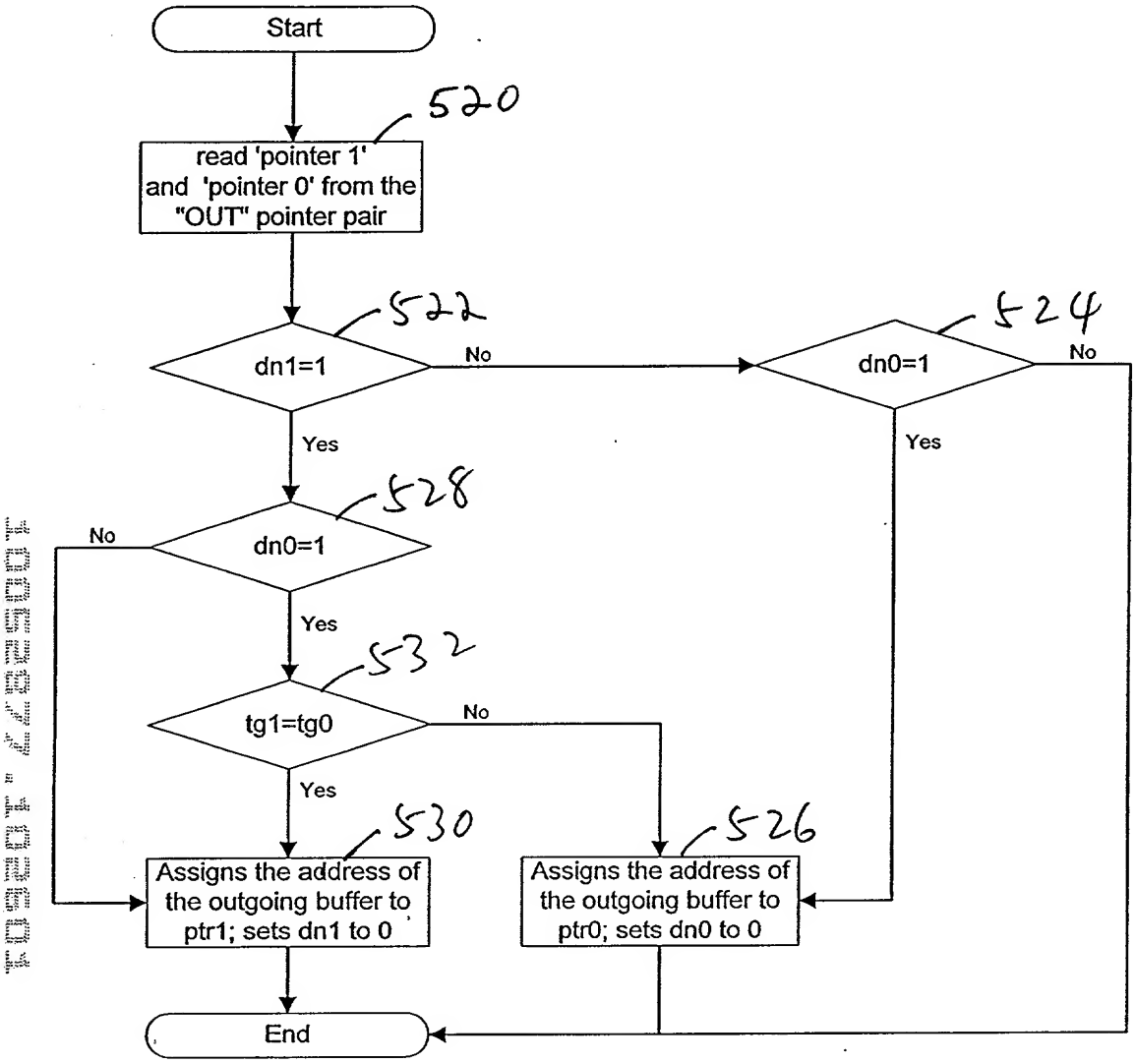


FIG. 40

case 2: An interrupt which is generated by the Packet Controller of the BT module indicates that an outgoing HCI packet is sent

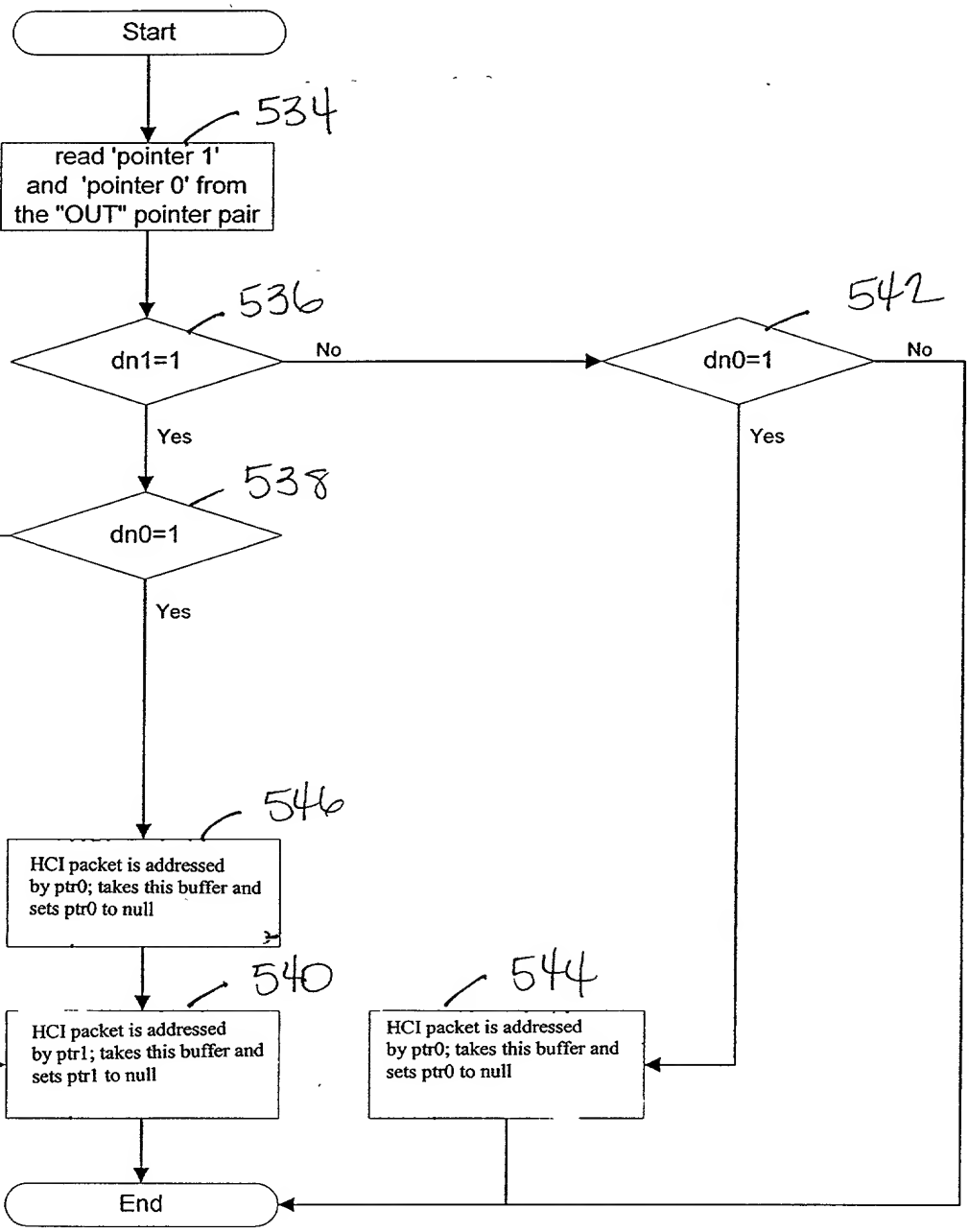


FIG. 4D

ACL RX precedure: Flow chat 1

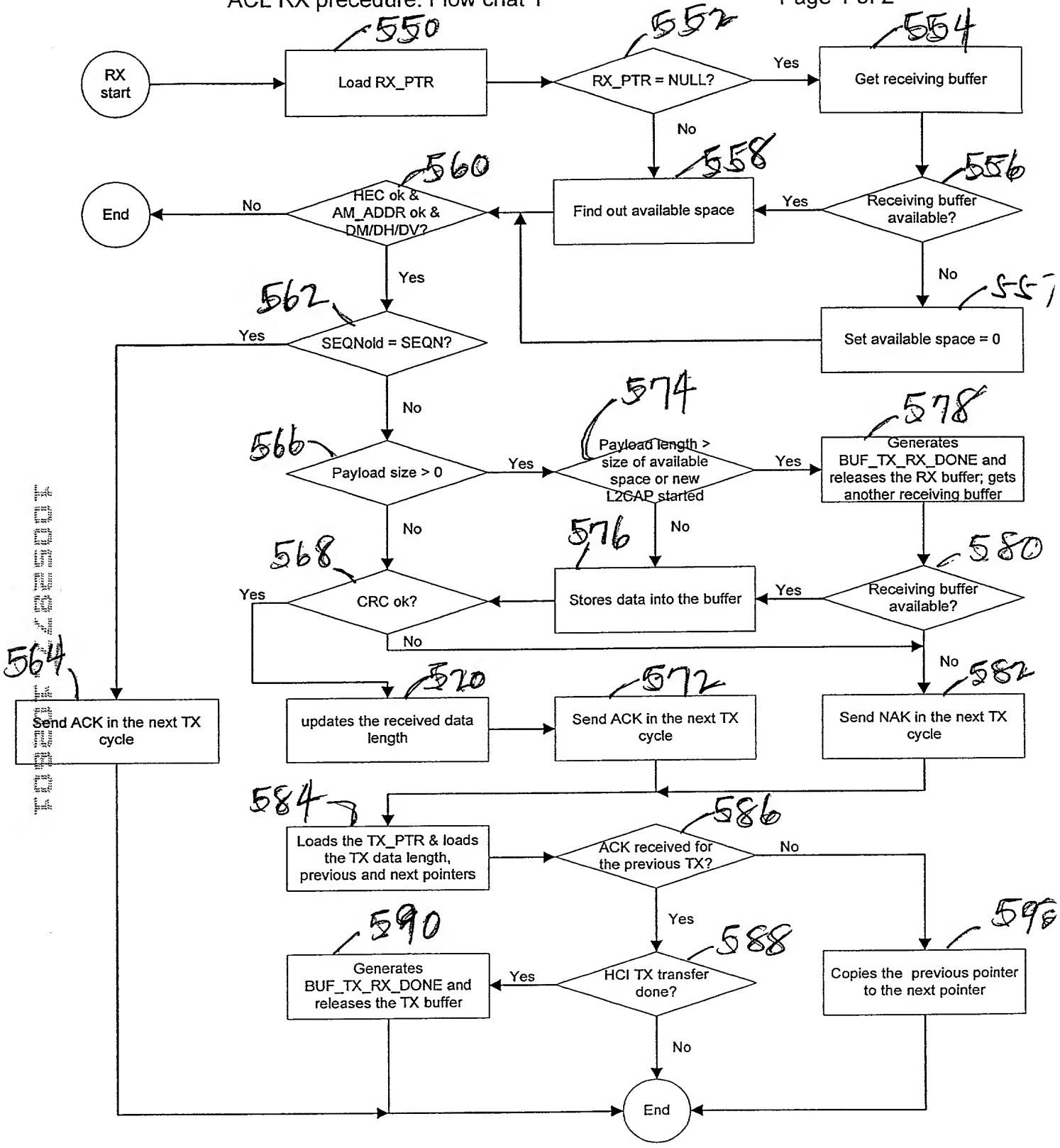


FIG. 42

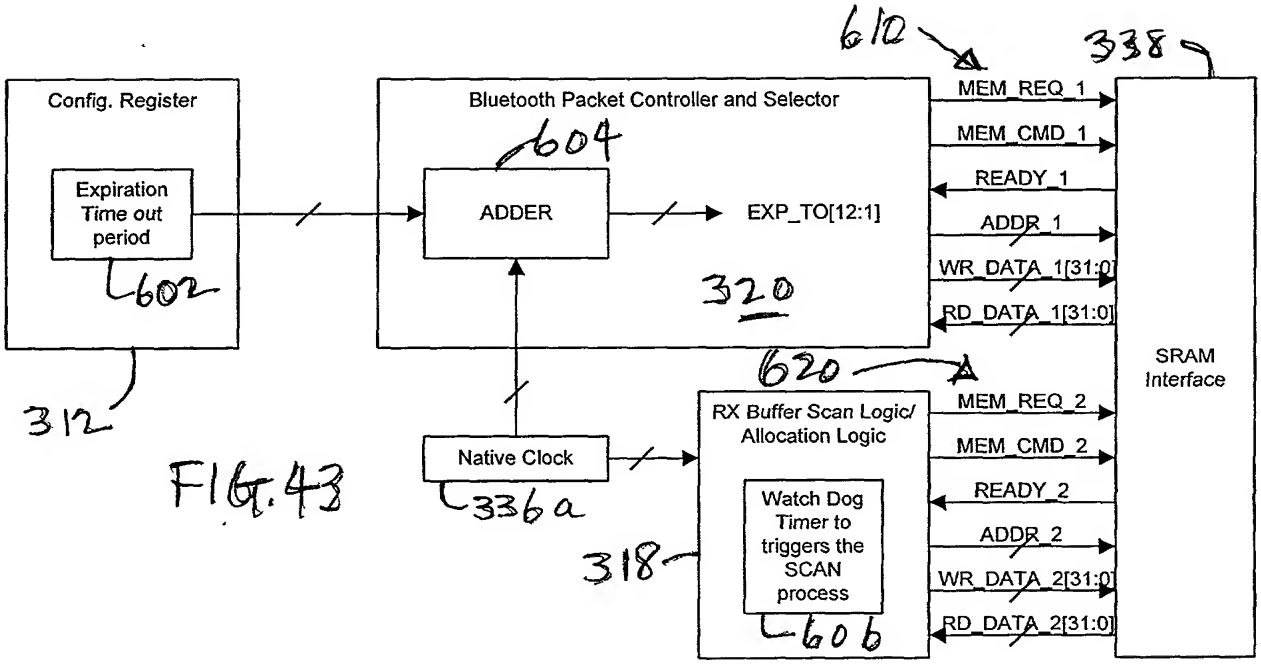


FIG. 43

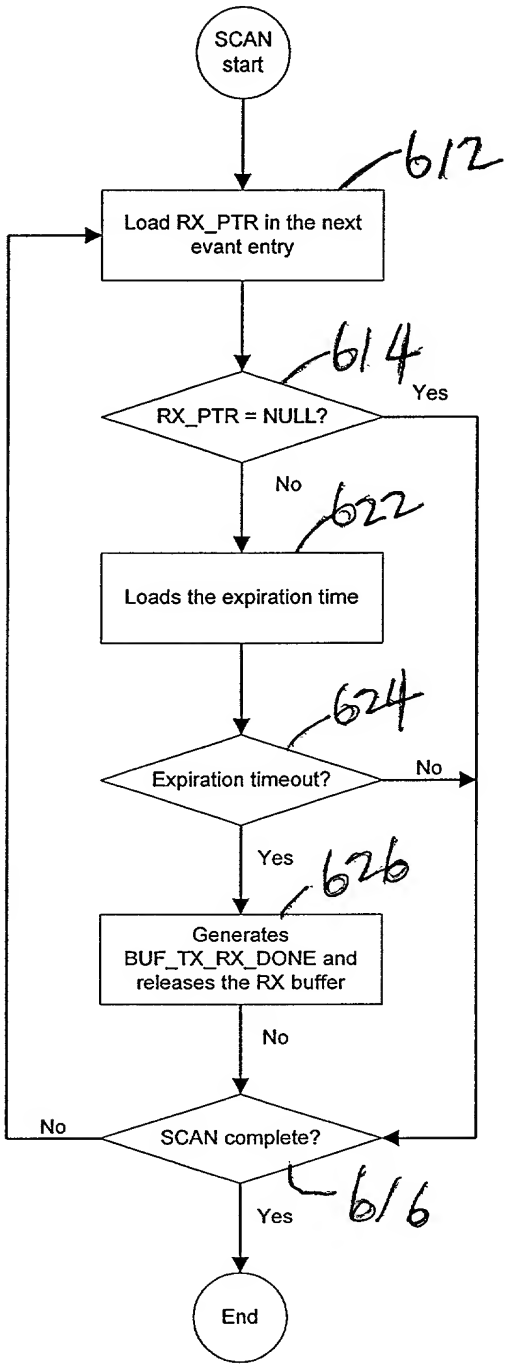


Fig. 44

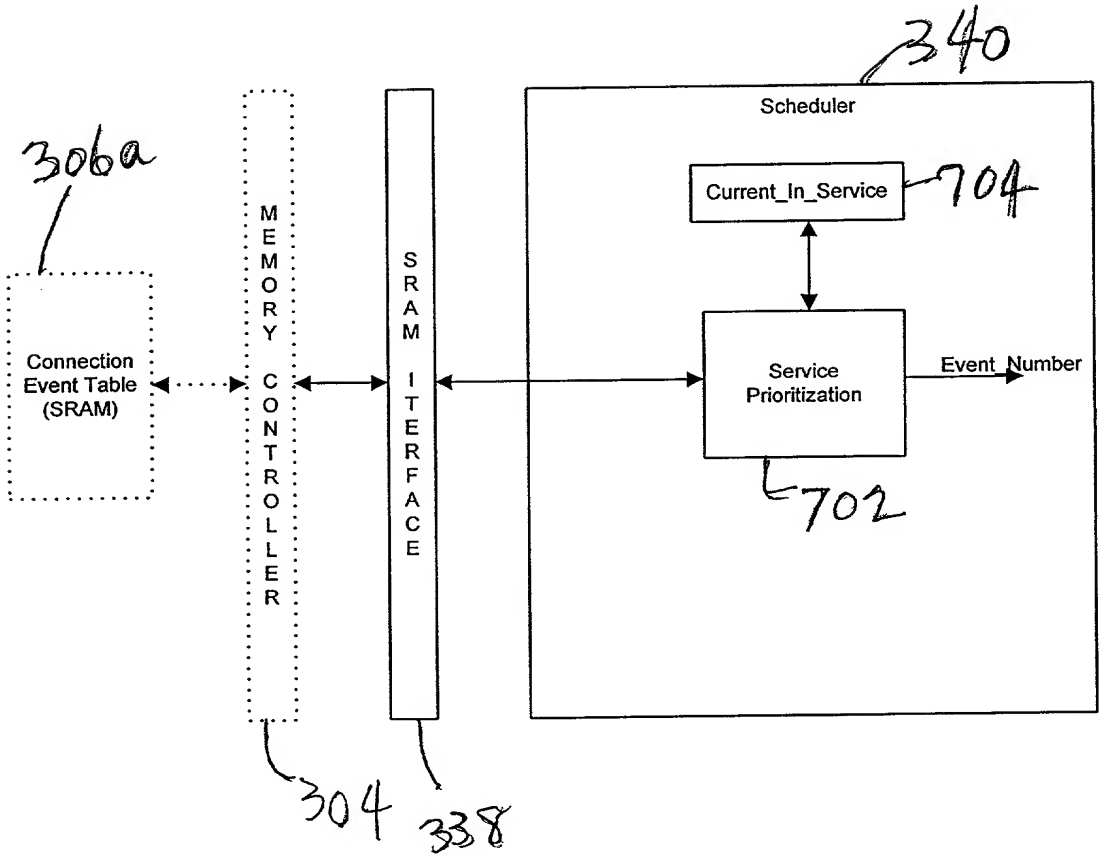


FIG. 45

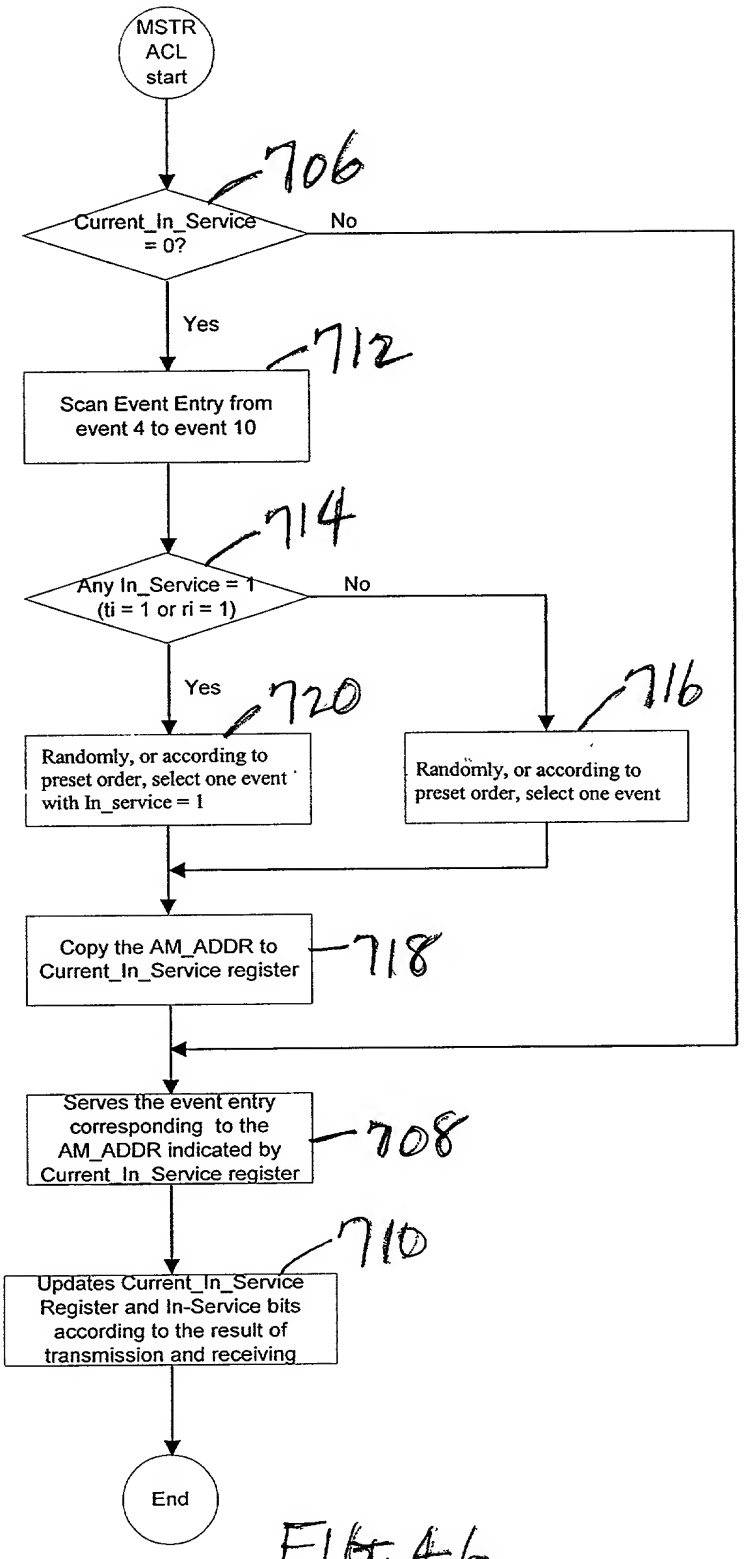


FIG. 46